The representation of engineering systems for the building, embodiment and optimisation with standard components

Hicks, B J

Award date:
2001

Awarding institution:
University of Bath

The representation of engineering systems for the building, embodiment and optimisation with standard components

Submitted by Ben Hicks

for the degree of Ph.D.

of the University of Bath

2001

Copyright

"Unfortunately for the majority they will only ever see the problems rather than the possibilities"

Hicks, August 2000

Summary

This body of research aims to improve support for engineering design and in particular the design of machine systems from standard components. It is widely accepted that the inclusion of standard components in systems can significantly reduce cost and improve the quality of design solutions, as well as reducing time to market. Standard components play an important role in engineering design, which like many other industry sectors is now a highly competitive global market.

To assist the incorporation of standard components in design solutions many suppliers and manufacturers produce electronic representations that govern the design and selection of a particular component. These representations are powerful tools for the identification and selection of individual components and significantly benefit the designer. However, the role of the designer is not just to select a range of suitable component sizes, but also to select an optimum mix of component types and sizes to deliver the desired performance characteristics, physical characteristics and quality at minimal cost. Current practices require the designer to manually evaluate many different combinations of component types and sizes in order to determine an acceptable solution. This iterative process is particularly time-consuming and analytically intensive. It is therefore highly desirable to support the designer over this process. To deliver such support a modelling approach is proposed which considers the system as a whole but also maintains the integrity of the various electronic representations necessary for the design and selection of each component. In this manner, systems of real components are dealt with.

A review of modelling approaches in engineering design and computer based support tools is undertaken. It is shown that current technologies do not provide for the modelling capabilities necessary to represent systems containing standard components. As a consequence, a new modelling approach is proposed that represents the performance of mechanical systems. In the development of the new modelling approach this research has had to address six key issues; system representation, a protocol for handling interactions, system resolution, data arbitration, compatibility analysis and interfacing third party electronic representations within the modelling approach. In addition to this, two other issues are investigated that are necessary for the strategic design of systems. These are cost forecasting for systems of standard components and the issues associated with the application of optimisation techniques. The feasibility of the overall modelling approach is demonstrated through the creation of a computer based support tool which is applied to a number of industrial case studies.

The research shows that it is possible to consider systems of standard components at the early stages of design and to provide for the automatic embodiment of conceptual solutions from standard components. Thereby freeing the designer's time to evaluate many more alternatives and develop a more refined design solution.

Acknowledgements

I would like to thank everyone who has supported my work and contributed towards this research over the last four years. In particular, I would like to thank Steve Culley for his constant support and encouragement as well as his keen eye for good diagrams. My thanks also extend to Glen Mullineux for his joint supervision and assistance during the programming phases of the work and to Tony Medland for providing financial support and advice over the last four years.

Special thanks go to all my colleagues over the period 1996-2001 who visited 4East 2.30. Even those whose visits were fleeting, you all played a part in this thesis. Whether it was introducing me to Quake (death match) or providing a never ending source of references. I would also like to thank many friends and colleagues in the Department of Mechanical Engineering who made the four years an enjoyable time, especially those who introduced me to golf and shared many a coffee in the Melting Pot.

Finally, a big thanks to my parents Peter and Lyn for their support, values and beliefs, without which I would never have had the opportunities presented to me in life so far, and in particular the chance to achieve this. And last, but by no means least, my thanks go to Dione for keeping me sane throughout the whole experience and for her constant love and support without which I could not have done it.

Definitions

Achievable designs in this context are those that are fundamentally based on existing technology and principles.

Agents are independent software objects that determine a feasible component specification from a set of performance requirements.

Bespoke components are elements that are precisely designed to meet a specific set of requirements and do not possess any predetermined sizes or properties other than those imposed by the design team due to resources, manufacture or production capabilities or other constraints specific to the particular application.

Binary elements possess two connections and in mechanical systems are the components which convey the inputs and outputs, or link the principal elements to other principal elements.

Complementary assessment provides for the qualitative considerations which the designer must undertake.

Components are mechanical items or the particular electronic representation assigned to each element of the system model.

Component attributes define the characteristics or properties of a particular mechanical component.

Component type represents a particular class of mechanical component from a specific manufacturer or supplier.

Connectivity analysis ensures that the geometric interfaces are matched and that energy interfaces are compatible.

Core components are the components from which all other component sequences emanate and provide the basis for all mechanical systems.

Electronic representations are the set of software modules that encapsulate a component based model, where this model represents the underlying principles or algorithms. The software modules provide the support functions necessary for manipulating the theoretical model in a software environment. They may provide for the visual interface, graphical support, search algorithms and user support functions, as well as access to the controls and object libraries contained within the operating system.

Elements are the individual parts of the system model.

Global attributes define system level performance attributes or physical attributes which are applicable to all the elements within a system.

Integrated modelling represents a system by manipulating and integrating the governing models for individual elements.

Intrinsic attributes are component attributes that are specific to a particular component type, range or size.

Local attributes are a subset of component attributes. Local attributes are component attributes that are determined in part, or full, from parameters that are driven by physical connections with other components.

Models represent a particular mechanical component and are termed component based models. These models are generally created from accepted scientific principles and provide for the sizing, selection and specification of an engineering component.

Parameters are physical quantities that may either explicitly define the value of an attribute or be used in combination to determine the value.

Performance matching ensures that the magnitudes of energy transfer are acceptable output and input levels for coupled components.

Performance modelling deals with the representation of performance. Where this performance represents a measure of the capability of the considered system to undertake or achieve its desired function. This approach truncates the conceptual and embodiment phases of the design process and aims to support the designer in rapidly embodying and testing various solutions. This embodiment determines a set of parameters for the system elements which meet the desired performance characteristics for the design.

Preferred designs utilise configurations and combinations of components that are deemed to be more suitable or better performing for the particular application.

Primary elements are those mechanical elements that provide the overall transmission requirements of the system.

Primary component attributes are the fundamental parameters upon which a component is specified or selected, and may include both physical and performance attributes.

Principal elements are the elements about which sequences of elements emanate and represent the core components in a mechanical system. Principal elements are identifiable by the characteristic that they possess more than two connections and are therefore always the core components in the system considered.

Real components are those elements which may already exist, have been previously used and have predetermined or predefined properties. In many cases, these are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified.

Resolution episode is the overall process of determining a solution state.

Secondary components are designed post selection of the primary components and may include housings and casings.

Secondary component attributes are more often than not either fixed values or follow a predetermined, discrete range of values for the considered component, such as the permissible angular misalignment of a bearing.

Standard components include both standard catalogued components and standard designed components.

System resolution involves the determination of a specification for a set of components that satisfy the given performance requirements and constraints for the system and individual components.

Systemic approaches considers a system and its included elements as a whole.

Unitary elements possess only a single connection and are the marginal or boundary elements to the system, and it is these elements which provide the system inputs and outputs.

Virtual agents are secondary software objects that act as intermediaries between the component representations and the system modelling environment.

Glossary of terms

AIE	Assembly Interface Entity
B-Rep	Boundary Representation
CAD	Computer Aided Design
$\mathbf{CAD_r}$	Computer Aided Drafting
$\mathbf{CAD_T}$	Conceptual Assembly Design Tool
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacture
CAPP	Computer Aided Process Planning
CE	Concurrent Engineering
CFD	Computational Fluid Dynamics
CNC	Computer Numerically Controlled
CSG	Constructive Solid Geometry
DDE	Dynamic Data Exchange
ERC	Electrical Rule Check
ERP	Enterprise Resource Planning
FEA	Finite Element Analysis
IDE	Integrated Design Environment
IGES	International Graphics Exchange Standard
IT	Information Technology
KBES	Knowledge Based Engineering Systems
MRP	Manufacture Resource Planning
NPI	New Product Introduction
OLE	Object Linking and Embedding

PC Personal Computer

PCB Printed Circuit Board

PDES Product Data Exchange Standard

PDS Product Design Specification

STEP STandard Exchange of Product data

VBA Visual Basic for Applications

Figures

Contents

Chapter 1

Introduction

The activity of engineering design itself has been applied to, or practised, in the creation of almost every artefact. Engineering design is widely understood to be the practice of transforming a set of perceived needs into an artefact, using creativity, scientific principles and technical knowledge (Evbuomwan *et al*, 1996; Konda *et al*, 1992). It is during this transformation process, termed the 'design process' that the properties of the artefact such as function, performance, safety, ergonomics, operation and maintenance are determined. In addition to these properties, the design process will have significant influence over the production costs and production lead times as well as dictating the necessary manufacturing procedures and level of quality, all of which determine the commercial success of the artefact. Because the design process plays such a critical role in determining the 'final cost' of the artefact, considerable attention is given to the various economic issues during the undertaking of the process.

In order to improve the practice of engineering design, both industry and academia invest considerable resources on researching new methods, techniques and technologies to support the designer in developing high quality, economical and technologically advanced solutions. A major focus of engineering design research is concerned with investigating and optimising the various activities and tasks of the traditional design process[1] and subsequent manufacturing process. The key areas for research in the field of engineering design theory and methodology are outlined by Finger and Dixon (1989a & 1989b) as: descriptive, prescriptive and computer-based models of the design process; languages, representations and environments for design analysis and support of design; design for manufacture and the product life cycle. This research has supported the development of philosophies and techniques such as Concurrent Engineering (CE) (Carter & Baker, 1992), Material Resource Planning (MRP) (McMahon & Browne, 1993), Enterprise

---

[1]The traditional design process is considered to be the systematic product design process described by many researchers such as Pahl & Beitz (1996), Pugh (1991) and Ullman (1992), and embodied in national standards BS7000 (1999) and VDI 2221 (1986).

Resource Planning (ERP) (Bancroft *et al*, 1998), Computer Aided Drafting (CADr), Computer Aided Manufacture (CAM) (Zeid, 1991), as well as numerous analytical techniques such as Computational Fluid Dynamics (CFD) (Anderson, 1995) and simulation tools. These are not meant to be exhaustive but merely illustrate some of the major advances in the areas associated with engineering design.

The majority of developments in the field of engineering design can be separated into two mutually supportive threads, defined by Law (1993) as the team-based approach and the technology based approach. The former aims to improve the <u>participation</u>, <u>integration</u>, <u>communication</u> and <u>organisation</u> between individuals from each of the departments that are involved in the new product introduction (NPI) process, and in particular design, manufacture, marketing and sales. The latter aims to <u>develop</u> and <u>improve</u> both new and existing <u>technologies</u> for the support of the tasks undertaken by individuals in the activities of design, manufacture, marketing, sales and distribution.

1.1 Computer support for the design process

The technology based approach highlighted in the previous section is particularly reliant upon computational support and significantly benefits from advances in the area of computer software and computer hardware[2]. In fact, Whitney (1990) highlights the computer as the main advance and driving force in engineering design today. A recent survey by Boston *et al* (1998) revealed that 95 percent of engineers have access to and rely on computers in order to undertake their respective design tasks. In addition to this, as long ago as 1993 it was estimated that over 80 percent of designers in UK manufacturing utilise computer aided drafting tools (Knutton, 1993). It is widely accepted that advances in computing capabilities have benefited nearly every industry, of which, most are now reliant on computers for undertaking their core trading activities as well as all the necessary support functions. In the field of engineering design, computers are certainly the primary enabling technology and their introduction has yielded a number of key benefits. These can be summarised as:

- The ability to produce results that were not previously obtainable with manual methods (McMahon & Browne, 1993).

[2] The team-based approach also benefits from the advances in computational techniques especially those which improve communication and decision making.

- Locate, retrieve and reuse data in a more efficient and effective manner (Pye, 1991).
- Transfer data rapidly between different users, departments or locations (Laudon & Laudon, 1998).
- Visualise and analyse complex geometries (Zeid, 1991).
- Automate routine and iterative tasks, such as searching and optimisation (Krick, 1969).

Many of these benefits are clearly exploited by software analysis tools such as Finite Element Analysis (FEA) (Fagan, 1992), Knowledge Based Engineering Systems (KBES) (UGS, 2001), Computational Fluid Dynamics (CFD) (Anderson, 1995) and Virtual Prototyping (Rosen *et al*, 1995). The realisation of Computer Aided Engineering (CAE) tools which deal with engineering analysis, detailed design, assembly, manufacturing and procurement, combined with more effective management practices has significantly reduced errors, improved the quality and the reliability of designs, as well as reducing product development costs and the time to market (Pye, 1991).

1.2 Computer support for the early phases of design

Although the number and diversity of commercial software packages for the support of engineering design continues to rise, many tools are limited to particular phases of the design process. Further to this, many of these computer based tools are only truly useful during the latter stages of the engineering design process as discussed in section 2.7, and as a result, supportive tools and methods for the early stages of product design, where most benefits can be accrued, are limited[3] (Baya & Leifer, 1995; Cartmell *et al*, 1993). This deficiency or lack of supportive methods is particularly evident at the transition from the conceptual to the embodiment stage of the design process, during which a concept is transformed and refined into a fully embodied design solution[4].

One of the most important requirements for support during the early phases of the design process is the supply of information to help the designer develop an embodied solution. The tools and methods that do provide for support during this critical phase of the design process generally

[3] The vast majority of the final cost of an artefact is obligated during the early phases of the design process, and is discussed in chapter 2.

[4] A review of supportive methods and computer based tools for the design process in chapter 2, highlights a deficiency in techniques that deal with the transformation of a concept to an embodied solution.

focus on the electronic[5] specification or selection of individual engineering components. The most common of these tools include electronic catalogues, parametric models, numerical codes, standard design procedures such as BS 436:1 (BS 436:1, 1967) and component libraries contained within the fourth generation CAD systems (SolidWorks Corporation, 1998).

1.3 Selecting and using standard components

In engineering design, competitive advantage arises to those companies that can produce competitive products. This demands that designers produce high quality, high performance design solutions, with both low development costs and low production costs. One approach to achieve high quality low-cost engineering is to utilise standard components.

Many authors: Pahl & Beitz (1996); Ulrich & Eppinger (2000); Harmer *et al* (1998) and Clews & Leonard (1985), discuss the importance of standard components. In fact, the origins of standard components, where this standardisation relates to the production of gauged or standardised component sizes for mass production, can be traced to the early 1800's when Brunel devised a mass production line for the manufacture of blocks for ship rigging, producing over 100,000 units in the first year (Model Engineer, 1993). Although, Henry Ford is often credited with implementing interchangeable (and by definition standardised) fixtures, fittings and associated standardised production techniques for mass production in the early 1900's (Womack *et al*, 1990).

However, the use of standardised component types and sizes for the configuration of different products, can really be attributed to Alfred Sloan of General Motors in the 1920s. The philosophy of Sloan was to standardise many mechanical items, such as pumps and generators across the entire product range (Womack *et al*, 1990), whereas Ford used standard parts to rationalise the components for a single product. The advantages yielded through the mass production of standard components for both the manufacturer of the component and the user were tremendous, and since these early years standardised components, and manufacturers of standard components have become commonplace.

[5] Electronic representations encompass any computational method for the design, specification or selection of a particular engineering component.

The benefits of standard components for today's engineer are well documented, and include:

- Increased quality and performance at decreased unit cost. This is typically because the third party producer can manufacture high volumes, thus reducing unit cost, and invest in learning and improvement of the component's design and production process, thereby improving quality and performance (Ulrich & Eppinger, 2000).
- Components have been historically proven and possess stable, known performance characteristics and reliability (Wallace, 1995).
- The use of standard components also yields some indirect benefits such as improved availability, global acceptance, established maintenance procedures as well as assured long term concurrency of components (Hicks & Culley, 2001).

The effective identification and utilisation of standard components can therefore significantly improve the quality and performance of a design solution. In fact, studies by Theobald (1995) have shown that standard components can provide between 50-80 percent of the components in a machine system. As global competition continues to rise the utilisation of standard components is likely to increase, because of the many advantages previously discussed. Consequently, methods that deal with the identification, selection and procurement of standard or third party engineering components is a very important research area.

There are two key areas of research associated with standard components, namely methods for the electronic representation and selection of engineering components and approaches which support the integration and participation of component manufacturers and suppliers in the design process. These two areas of research have been instrumental in driving the development of electronic catalogues (Culley & Webber, 1992), the establishment of common practices for the geometric representation of engineering components such as STEP (Pierra, 1994); the generation of part libraries for fourth generation CAD systems (Autodesk Inc, 1997), the development of information systems for stock control, availability and pricing (Curtis, 1991) and methods for assessing and liasing with suppliers (Boston *et al*, 1999). All of which provide support for the designer during the <u>identification</u>, <u>selection</u>, <u>procurement</u> and <u>assembly</u> of standard or third party engineering components.

The adoption of these technologies over the traditional manual based processes for engineering component procurement[6] yields distinct benefits for the designer, these are documented by a number of authors: Vogwell and Culley, 1991; Reinemuth, 1993 and Webber, 1994. The main benefits can be summarised as:

- The removal of a lot of the time consuming searching, calculations and analysis.
- The ability to consider many more individual component options from different manufacturers.
- The provision of accurate and complete information for current products.
- Up-to-date information on cost, stock, ordering and availability from the manufacturer or supplier.
- The ability to design strategically from suppliers and optimise component selection, for example costs.
- The provision for automatic inclusion and importing of two-dimensional and three-dimensional computer models.

1.4 Standard components and system design

Much of the emerging research and new technologies, and in particular electronic catalogues, deliver significant advantages over the manual procedures for the identification and selection of engineering components. However, most approaches consider components in isolation, where as engineering assemblies are generally complex systems (Pahl & Beitz, 1996; Flood & Carson, 1988), and as such should be considered in a holistic manner. In particular, if an overall optimum is to be established then a systemic[7] approach that considers standard components and their associated representations is needed. Through evaluation of current technologies, four key limitations to their ability to consider systems and standard components during the early stages of the design process are identified:

[6] For the purpose of this work, procurement encompasses all the necessary tasks to identify, size or select, specify and acquire a particular engineering component from a third party supplier.

[7] For the purpose of this work, a systemic approach is one which considers a system and its included elements as a whole.

- As true globalisation becomes more of a reality, the designer has access to many more component manufacturers and suppliers, each providing many different types of component which are available over a large number of discrete sizes. This makes for an exponential number of possible configurations for component types, sizes and combinations, which must be investigated by the designer. Unfortunately, many suppliers have adopted their own ad hoc standards for their electronic selection procedures, requiring the designer to be familiar with all the intricacies of operation for the various search techniques, controls and interrogation procedures for different component types.
- The wealth of emerging electronic media combined with its rapid development, has resulted in many different software environments, such as web-based applications, CD-ROMs, spreadsheet systems and numerous CAD environments. This demands that the designer be familiar with all these different applications, and frequently different platforms.
- The current technology and tools for the selection of engineering components only provide for the selection of <u>individual</u> components in isolation. The utilisation of standard components in the context of systems design is largely overlooked by many authors (Webber, 1994; Reinemuth, 1993). A systems approach is very important because of the highly coupled nature of mechanical components, which must be represented in order to develop a feasible and optimal solution.
- In the pursuit of an optimal design solution it is essential that the system of components is considered as a whole, where this consideration is provided within a single controlling software environment. Consequently, if optimisation of systems comprising standard components is to be achieved then the various representations for individual components must be interfaced or integrated with optimisation software. This need for integration or interfacing of computer based systems was identified as an important area for future research and development by a recent industrial workshop (Culley, 1999).

Whilst the diversity of component selection procedures and representations does yield inherent limitations, this diversity has certainly facilitated the progression and advancement of electronic selection procedures. Consequently, many of the current electronic representations provide very powerful design and selection tools for individual components. Therefore, it is desirable that the integrity of these representations be maintained, so that the benefits of each class or type of representation can be realised within a systems approach. In addition to this, it is essential that real[8] engineering components are considered at the early stages of design and also during optimisation.

1.5 Research aims

The previous sections highlight the significant benefits and the important role of standard components for machine systems design, and argue that their utilisation is going to rise in the future. These factors, coupled with the desire of the designer to develop the best solution, demand that techniques which deal with strategic and optimal design of machine systems also consider standard components. Furthermore, because the influence of standard components on the success of the design is so significant, and the majority of costs are obligated during the early stages of the design process. It is particularly important for procedures that size, select or specify standard components to be considered during the early stages of the design process. In order to achieve this, the ability to model and analyse design concepts for their embodiment with standard components is needed. Such an approach would enable the consideration of standard components at the earliest possible stage in the design process.

To address these issues three hypotheses are proposed:

Hypothesis 1

The electronic representations for standard engineering components can be manipulated in such a manner so as to enable the performance of mechanical systems to be represented.

Hypothesis 2

This approach can be implemented in a computer based support tool to enable the representation of topology and performance for conceptual systems of standard components.

Hypothesis 3

The approach can be extended to enable the configuration, embodiment and optimisation of engineering systems from standard components.

---

[8] The term 'real' denotes those elements which may already exist, have been previously used and have predetermined or predefined properties. In many cases, these are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified.

In order to investigate these hypotheses a number of key objectives have been identified:

- Investigate modelling approaches for engineering systems in other domains, with a view to the development of a strategy for representing machine systems and in particular rotating power transmission systems.
- Develop a modelling approach for machine systems that represents the performance capabilities of the system and individual components.
- Investigate the various classes of electronic representation for engineering components and develop a methodology for their consideration within a machine system model.
- Create a computer based support tool for the configuration and embodiment of mechanical systems from standard components.
- Identify the requirements for the optimisation of mechanical systems configured from standard mechanical components, and develop the issues for the parametric optimisation of a machine system model.
- Demonstrate the modelling approach through the application of the computer based tool to industrial case studies.

These objectives are addressed by the work described in each of the chapters.

1.6 Thesis structure

This chapter has provided a brief introduction to engineering design, the economic and global issues which drive the need for improvements in the design process, the range of industrial and academic research and their deliverables. The importance of standard components, their current utilisation in engineering design and their considerable benefits are highlighted. Furthermore, the lack of supportive tools and methods for the important area of <u>systems</u> <u>design</u> from <u>standard components</u> is discussed, and the limitations of current technologies and techniques for representing the performance and geometry of mechanical systems are described. These limitations or deficiencies drive the need for research into methods that deal with system design, and the development of techniques that consider the various emerging and current technologies for representing individual engineering components within a systems approach. The following paragraphs provide a synopsis of each chapter, and an overview of the thesis is given in figure 1.1.

Chapter 2 provides a detailed review of engineering design and relevant research work. In particular, the review focuses on the traditional engineering design process, and the various types of design activity undertaken by engineers today. This highlights the importance of standard components in the design of systems and the reliance of today's designer on computer based support tools. Further to this, the importance and advantages of electronic selection procedures are described and their limitations with respect to the evaluation of performance and geometry for systems is discussed. This chapter concludes by identifying and developing the outstanding research issues that form the basis of the hypotheses of this work.

Chapter 3 reviews systems modelling approaches in other engineering domains, and in particular fluid power and electrical circuit design. Following this, a review of standards and techniques for representing data describing engineering components and/or their interactions is undertaken, and the diversity of electronic (computer based) representations for individual engineering components is discussed.

Chapter 4 describes the development of a system modelling approach for mechanical systems, and in particular machine systems. This includes system representation, representing interactions and resolving the system model, which are necessary to achieve a flexible[9] component based[10] modelling environment. This chapter also introduces the key aspects and architecture necessary for an integrated modelling environment that provides for the embodiment of mechanical systems from standard selected and standard designed components.

Chapter 5 and chapter 6 build on the modelling approach developed in chapter 4. Each chapter discusses the development of the functions necessary to support system embodiment and in particular data arbitration and compatibility analysis. The requirements for each of these functions are discussed and strategies which compliment the proposed modelling methodology are developed. These support functions ensure that a system of components is determined which is free from conflicts and ambiguities, and is capable of delivering the required performance.

Chapter 7 categorises the various types of electronic representation for mechanical components, and discusses the importance of integrating or interfacing these current and emerging

---

[9] Flexible is used in the context of this work for approaches where there are no prescribed methodologies or procedures for the modelling of mechanical components *per se*.

[10] Component based is used to denote an approach that considers individual components and their associated representations.

representations with a single design environment. A review of previous work in this area is undertaken and a number of approaches are investigated. The chapter concludes with an overview of a generic approach for interfacing electronic representations which is incorporated in the modelling environment developed in this work.

Chapter 8 provides an overview of a computer based modelling tool which implements the various aspects of the modelling approach developed in the previous chapters. An overview of the software architecture is provided and the software modules that provide for each of the key aspects of the modelling approach are described. Furthermore, a case study is used to illustrate the application of the software as a design support tool.

Chapter 9 discusses the need for multi-objective optimisation and strategic design. This identifies the need for complete component information at the early stages of design, however, attributes such as cost and mass are often absent. Consequently, methods that allow for the modelling of incomplete design data, and in particular cost, are developed for the full range of mechanical components.

Chapter 10 discusses the issues associated with the optimisation of systems of standard components. These are used to develop the requirements for a strategy that provides for the multi-objective optimisation of design solutions. This strategy deals with both discrete and continuous solution spaces, necessary in order to include standard components in a design and optimisation environment.

Chapter 11 describes the application of the modelling tool to a number of industrial case studies. This demonstrates the feasibility of the modelling approach and highlights the capabilities of the new approach and the significant benefits for the designer.

Chapter 12 reflects on the original hypotheses of this research and critically appraises the work. From this, a number of key conclusions are drawn and the contribution of the work discussed. Furthermore, a number of future research issues in engineering design are highlighted and possible directions for future work that addresses these issues are discussed.

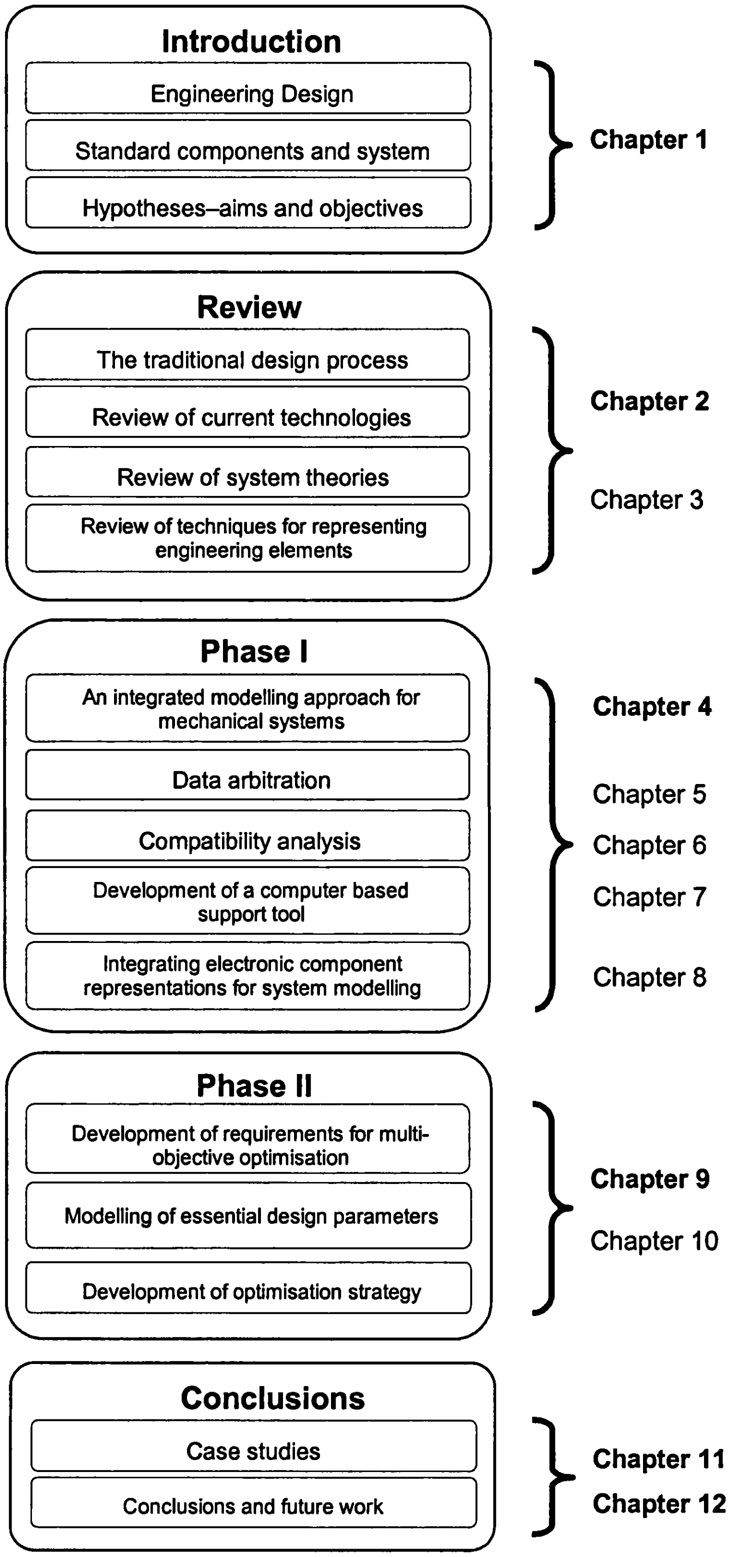


Figure 1.1 – Thesis structure

Chapter 2

Review of Engineering Design

This chapter provides an overview of engineering design, covering fundamental methods, processes and theory, as well as highlighting key technologies and research relevant to this work. Firstly, the importance of research in engineering design is discussed and models of the design process are described. The three classes of design activity: adaptive, variant and original (Pahl & Beitz, 1996) are discussed and their extents of practice in various engineering sectors are highlighted. Secondly, various modelling analysis activities which support the design process are described and their limitations discussed with respect to systems modelling and the selection of standard components. Following this, an extensive review of computer based support tools for engineering design is undertaken and the extents of applicability or usefulness for each tool over the design process is highlighted.

The major focus of engineering design research is concerned with investigating and optimising the various activities of the design and manufacturing processes. Figure 2.1 illustrates the typical product life cycle from which it is clear that ultimately effective design is a pre-requisite for effective manufacture. This investigation and optimisation aims to improve the quality, efficiency and effectiveness of both the aforementioned processes in order to "*better match products to customers needs*" (NRC, 1991), and develop a commercially successful product. These goals are addressed through intensive research by academia and industry which deals with:

- *Improved product quality*. This is measured in terms of the ideal quality a customer expects, i.e. that every product delivers the target performance each time the product is used, under all intended operating conditions and throughout its intended life and with no harmful side effects (Phadke, 1989).
- *Incorporation of advanced technology*. This requires the committal of considerable research and development funds, as much as ten to twelve percent of annual turnover, and generally yields technological leadership and ultimately products with superior functional performance (Holt, 1991).

- *Reducing time to market.* This requires the minimisation of the product development time and if achieved in relation to market competitors creates invaluable lead time for the product, as well as extending the product's saleable life. Therefore, a delayed or prolonged development time for a product can have a significant affect on profitability, which can be far greater than any increase in development costs necessary to reduce time to market (Smith & Reinertsen, 1995).
- *Reducing product cost.* The final product cost is determined by the various activities of manufacture, assembly, marketing, distribution, research and development, as well as operational costs for the respective company (Phadke, 1989). These costs are passed onto the customer and in conjunction with the market forces determine a product's saleable value and therefore its realisable contribution and profit.

The areas detailed above are not meant to be collectively exhaustive and are considered broad enough to encapsulate any traditional concepts which the reader may desire to include such as reliability, ergonomics and design for X type activities (Zeid, 1991). It is also noted that other factors such as management, marketing and technological strategies play a vital role in the quality, functionality and ultimately the commercial success of the product.

The development and application of research findings is essential for the advancement and improvement of engineering design *per se*. In the main, research findings are embodied and incorporated into the design process through either computer based support tools or techniques (Information Technology), design engineering (management) philosophy, or process models and methodologies, each of which may be tailored to a specific industry sector or a particular activity[1] of the design process. The key areas of work which impact on this research are the design process, the current state of modelling and analysis for systems design, the role of standard components in engineering design and computer based support tools for the design process, and in particular the early stages.

2.1 The design process

To improve the efficiency of engineering design, researchers have over the years classified and modelled the overall engineering design process in order to capture the special skills and underlying systematic thought and procedure. However, it is not meant to replace or remove the intuitive, creative and experience driven elements of design. The goals of such design methods

[1] These activities include any of the individual tasks of the design process as well as any class of design activity, such as redesign or adaptation.

are therefore to; encourage a problem-directed approach, foster inventiveness and understanding to facilitate the search for the optimum solution and the application of known solutions to related tasks, as well as ensuring compatibility with modern management-science and the concepts and methodologies of other disciplines (Pahl & Beitz, 1984). The design process is modelled by many authors as a systematic process which the practising designer may follow in a step-by-step manner, although iterative loops are prevalent throughout the various schemes. The typical or accepted models include those derived by Pahl and Beitz (1984), Ullman (1992) and Pugh (1991), depicted in figures 2.2, 2.3 and 2.4 respectively. These models aim to guide the designer to possible solutions and ultimately the most promising thereof more directly. The correlation between proposed systematic design models is strong and the systematic design process is generally understood to comprise four main activities; *clarification of the task* or *specification*, *conceptual design*, *embodiment design and decision making* and *detailed design*. These are described in an approximately chronological order due to their inherent dependency on one another.

2.1.1 Clarification of the task or specification

Clarification of the task is undertaken by the designer or design team with the appropriate parties. It primarily entails the identification and collation of the requirements which the customer desires in the final designed artefact. It also involves the development of constraints imposed by the customer on factors such as technology, availability of materials or particular components, consumables and the environment. Aspects such as ergonomics, aesthetics and safety may also be considered at this stage. This activity culminates with the production of a detailed specification that covers the desired functional, physical and performance characteristics, against which possible design solutions can be evaluated.

2.1.2 Conceptual design

The conceptual phase of the design process is described in many texts and often commands considerable explanation (Pugh, 1991). Authors attach great importance to this phase and describe many sub-processes in the overall activity. These include function structures, developing concepts for each function (functional morphology) (Ullman, 1992); identifying the solution principles and combining them into concept variants (Pahl & Beitz, 1984) and establishing a concept comparison and evaluation matrix based on a criteria generated from the product design specification (PDS) (Pugh, 1991). During this phase of the design process, it is essential that the designer continually seeks to combine the elements of strong concept variants in order to configure improved concept hybrids and ultimately a more effective design solution. The designer may need to construct detailed sketches and undertake preliminary analysis of the most promising concepts in order to evaluate and compare possible solutions against the desired criteria.

The majority of literature acknowledges the fact that each step in the conceptual phase must be intensely completed in order to achieve the best concept variant. Processes and tasks within the conceptual phase may be repeatedly performed and indeed steps four to six of figure 2.2 afford the possibility of unlimited iterations in order to develop the most promising solution (Pahl & Beitz, 1984). Indeed Pugh (1991) states that the included tasks should be rigorously adhered to and encourages as much iteration and refinement of designs as is possible. This is because it is much less expensive to extend the conceptual design phase than correct or refine a design later in the process. In fact, to correct the fundamental shortcomings of concepts in the subsequent embodiment and detail design phases it is extremely difficult or often impossible (Pahl & Beitz, 1996). It is also critical at the conceptual stage, that the designer seek as much input and information about new and existing technology from both formal and informal sources in order to develop the most successful design.

2.1.3 Embodiment design and decision making

During the embodiment phase of the process the designer or design team are concerned with determining the layout and form of the concept(s). The technical specification of the solution(s) must be determined in accordance with accepted design and manufacturing practices, these may be recognised standards or company specific procedures. The embodiment phase will typical involve tasks such as:

- The generation of scaled drawings for the solution.
- Sizing and determination of standard designed or standard selected components, such as shafts, keyways and chain drives or gears.
- Design of non-standard elements such as housings and brackets or even bearings.
- Evaluation of system characteristics and attributes to meet essential performance requirements.
- The consideration and inclusion of standard parts/components. The benefits of which are discussed in section 2.3.
- The rationalisation of system elements. This involves introducing interchangeable components into an assembly which reduces the spread of stock required by the manufacturer

and reduces cost. Furthermore, benefits such as improved reliability and maintainability can also be realised.

The final embodied solution(s) provides the basis for determining the functional, spatial and financial viability of a design, other measures may also be considered, such as strength, reliability and ease of manufacture.

2.1.4 Detailed design

This phase of the design process is sometimes combined with embodiment design because the distinction between certain activities is fuzzy or activities may be repeated in both phases. However, their emphasis or focus may differ, one such example is in the design of non-standard components. In the embodiment phase the designer is concerned with satisfying essential geometry and performance levels, where as in the detailed design phase the designer may be concerned with design for manufacture, material and assembly, as well as such tasks as optimisation of mass, power transmissibility or any predetermined criteria which the design is to be optimised against.

This phase ultimately culminates with the production of arrangement and part drawings for all elements of the solution. These specify arrangement, connections, form, dimensions, tolerances and the surface finish of each element, as well as listings of all standard/third party elements.

2.1.5 Bespoke models and methodologies

In addition to the development of process models and methodologies that support the overall design engineering approach. Design processes are often adapted for specific engineering design procedures in different industry sectors or specialist branches (VDI 2221, 1986). In fact, the adoption or generation of company specific methods are often cited as reasons for sustaining competitive advantage.

The main advantages of customised or adapted design methods include the focusing of design efforts to particular activities, the identification of essential activities and often the establishment of suitable tools and techniques in support of the specific activities. These are particularly important in heavy engineering where safety is important, or in machine tool construction where accuracy, speed and flexibility are essential, or the motor industry where appearance, ergonomics and cost are important.

These methodologies can be top down holistic approaches or bottom up methodologies derived from specific cases, such as the routine design of mixing machines (Brinkop *et al*, 1995) or a redesign methodology for packaging machinery (Hicks *et al,* 2000) respectively. In general design activities are separated into three classes: original, adaptive and variant.

2.1.6 Types of design activity

Numerous researchers in the field of engineering design such as Pahl and Beitz (1984), Ullman (1992), Pugh (1991), French (1988), Ulrich (1995) and Qian & Gero (1996) define different types of design activity. The various classes of activity and the definitions proposed by these authors are summarised in figure 2.5. Evaluation of the various definitions of design activity reveal similarities between different definitions and highlight certain activities as subclasses or combinations of one or more other activities. If these subclasses are combined and duplicates removed then three types of primary design activity emerge, each of which is summarised below.

1 Original or creative design

This class of design involves elaborating or developing an original solution principle for a process, component, plant machine or assembly not previously in existence. Although the task may be the same, similar or new, where the term task encompasses the required function of the solution. Ullman (1992) states that original design may not be reduced to any algorithm(s), each one represents something new and unique. Original design is sometimes classified as non-routine or creative design (Qian & Gero, 1996).

2 Adaptive, non-routine design or redesign

This type of design activity involves adapting a known system, or the modification of an existing product (the solution principle remaining the same) to a changed task. Here original design of various parts or assemblies is often called for.

3 Variant or routine design

Routine design is taken to mean that class of designing where all the design or structure variables and all the performance or behaviour variables are known *a priori* and what is to be done is to determine values for the structure variables. This involves varying the size and/or arrangement of certain aspects of the chosen system, the function and solution principle remaining unchanged. No new problems arise as a result of, say, changes in materials, constraints or technological factors. When all these variables are not known at the outset we move into an area characterised as non-routine or adaptive design (Qian & Gero, 1996).

Activities such as dynamic or non-evolutionary design, and static or evolutionary design are merely a combination of original and adaptive or adaptive and variant design respectively, shown in figure 2.6. Indeed, the classification of a design project or task into a class of design activity depends on whether the design specification relates to a system, an assembly or even a subassembly. For example, a product may be of dynamic design but its individual subassemblies may be either adaptive, variant or original, or a combination thereof.

Many authors have undertaken studies investigating the division or spread of design activities in engineering industry. These include Court (1995); Pahl and Beitz (1984); and Pugh (1991). Pugh (1991) suggests that 80 percent of typical design is adaptive whilst a survey by Pahl and Beitz (1984) suggests that 55 percent of products are based on adaptive design, 25 percent on original and the rest on variant. Although there is a substantial variation in findings between the studies, both demonstrate the fact that adaptive and variant design describe the majority of today's design projects. In undertaking adaptive or variant design tasks, the designer will typically re-specify a number of machine elements in order to deliver the required changes in performance capabilities. For the task of re-specification or resizing of components standard components play an important role and enable the designer to rapidly undertake such activities (Harmer *et al*, 1998; Ullman, 1992). However, few authors discuss the important task or activity of selecting standard components. The role of standard components is discussed in detail in section 2.3.

In contrast to the discrimination of design activity, some authors such as Hubka (1982) and Hales (1993) do not distinguish between the types of design activity and propose design process models that are neutral with respect to the artefact being designed. Indeed Hubka (1988) describes a model of the design process that may be applied to the design of all kinds of machine systems or technical systems. In fact, none of these authors or standards delineate between types of design activity within their process models. These models are exhaustive and encompass every conceivable element of the design process. However, such high level process models do not provide a focus on certain tasks, for say adaptive design rather than original design. In fact, a limitation of many of the models of the design process is that they fail to direct the designer to support tools, techniques, information sources or any other aids which may benefit the particular design activity, design problem or a particular stage of the design process. Because of this, a review of various modelling approaches is undertaken and their extents of applicability over the design process highlighted. In particular, this review aims to identify tools which support the transition from concept to embodiment and those which deal with standard components and may therefore impact on this work.

2.2 Modelling activities in engineering design

The reliance of the design process on modelling, combined with the emergence of advanced numerical analysis and computer based representations, has resulted in a range of modelling and analysis activities which may provide for many of the core tasks in the design process. However, no single modelling or analysis activity when used in isolation can provide for the full range of tasks necessary for total design and manufacture. Although for particular classes of design activity, such as adaptive or variant, these modelling approaches can provide for the tasks necessary to complete the majority of phases within the truncated design process.

Through the evaluation of a range of modelling activities, their relative extents of application over the design process can be represented, shown in figure 2.7. The classes of activity dealt with are not meant to be exhaustive but encompass the principal modelling approaches available for representing function, geometry, behaviour and performance of engineering systems. For this work, the following definitions are adopted.

- *Function* deals with the functional properties or purpose of a component or system. These properties are generally described using a natural language formulation.
- *Geometry* deals with the physical dimensions of each component as well as the relative arrangement or topology of the system. Geometry may also describe the interfaces between one or more elements.
- *Behaviour* describes the manner by which the function of a component or system is achieved.
- *Performance* is a measure of the capabilities of a component or system to deliver its function.

The modelling activities discussed in the next sections can be considered to encompass many of the current and emerging computational tools for the support of the design process. Definitions of each modelling activity are provided and their role in the design process discussed.

2.2.1 Geometric modelling

Geometric modelling is generally considered to deal with three-dimensional representations of an object or a collection of objects which constitute an assembly. The methods that have been developed for three-dimensional modelling involve the representation of geometry as a collection of lines and curves (wire frame), surfaces or solids in space. The advantage of three-dimensional modelling is that a single representation is used to generate various instances, perhaps for different views, without the risk of inherent errors often generated when projecting two-dimensional representations into planar space (McMahon & Browne, 1993). The three methods for generating three-dimensional models give rise to three distinct categories of representation: wire frame, surface modelling and solid modelling.

- Wire frame modelling represents the geometric edges of an object as a combination of lines, arcs and splines arranged in three-dimensional space. For complex objects, the amount of information that a wire frame model contains is as likely to be a source of confusion as of clarification (Medland & Burnett, 1986).
- Surface modelling represents the boundaries of an object and forms surface meshes between these boundaries. There are a range of meshes or surface entities available and it is important to select the correct one for the particular application. Some common surface entities include plane surfaces, Bezier surface and B-spline surfaces (Zeid, 1991).
- Solid modelling is the representation of an object as a solid (a space totally bounded by a surface). There are two methods for constructing solids; boundary representation (B-Rep) and constructive solid geometry (CSG). The former generates a solid by sweeping a model space with either a line or a wire frame about a particular axis and retains the enclosed space as a solid. The latter approach generates a solid from primitive solids which are combined using Boolean operators to achieve the desired form.

The objective of geometric modelling can be to visualise, inspect or evaluate an assembly or arrangement in isolation or in virtual surroundings. This is important for ergonomics, style and aesthetics. Computer based tools such as Jack (EAI, 2000) use virtual environments to evaluate the functionality and ergonomics of products, one such example is the field of vision that a driver would have in a new vehicle design. In addition to this, limited static analysis can be performed such as geometric fits, interference and mass calculations. Furthermore, surface modelling is essential for developing finite element models for detailed analysis during embodiment design and can play a vital role in generating tool paths for Computer Numerically Controlled (CNC) machining (Medland & Burnett, 1986).

2.2.2 Parametric modelling

Parametric modelling describes a component or assembly in terms of parameters and constraints. The goal is to assign values to parameters so that no constraint is violated and all design requirements are satisfied. Here the design requirements are transformed from functional descriptors into geometric, physical or other parameters which pertain to the component(s). In parametric design, the design state space or solution space is determined by the number of parameters (degrees of freedom) which the designer may vary. In the case of components or assemblies that are governed by a set of equations and rules, the solution space can be very large, in which case it is often necessary to introduce heuristics or other methods in order to reduce the solution space and determine a solution. Ullman (1992) illustrates one such example, where the requirement is to design a tank that holds $4m^3$ of liquid. If the height and diameter of the tank are variable then there is an infinite number of solutions, so in order to determine a solution robust design methods are used. Through the consideration of manufacturing tolerances an optimum set of parameters for the diameter and height can be determined.

Parametric design may be considered a subclass of configuration design where the interconnection of components is given in advance and the problem is only to select components and calculate the values of their parameters in order to meet the design requirements. Further to this, constraint modelling is often considered to be an extended and more flexible approach to parametric design

2.2.3 Constraint modelling

Constraint modelling is defined by Bahler *et al* (1990) as a formalisation that represents mutually constraining parameters and their relationships as a network of inter-related constraints. Constraint modelling techniques for design aim to represent what is to be achieved, typically performance requirements, rather than how it is to be achieved, in terms of process, which parametric modelling will typically implement (Medland, 1990). These goals are represented as constraint rules between design parameters which may be evaluated at any stage in the process. The aim is to find a solution that satisfies all the imposed constraints. The solution space is the intersection of all the individual constraint fields. This shows all possible solutions as well as those solutions that fail. For many constraint modelling environments this intersection is determined by direct search techniques which minimises the error for the given set of constraints, and converges on a successful solution or the best compromise solution. In this manner, a constraint approach allows changes in both the proposed solution and in the constraints. The former through a direct search approach and the latter by changes in strategy.

2.2.4 Functional modelling

Functional modelling is described by Rosenman and Gero (1997) as the modelling of functional properties of a designed object at an assembly or component level. This definition does not formalise the generation and content of a functional description for an artefact, but merely generalises. This is not just a shortfall of Rosenman and Gero but many authors who detail functional modelling as the relationship between inputs and outputs of an artefact (Pahl & Beitz,

1996) or the intended use of the artefact (Ullman, 1993). In fact, there are two classes of model which fulfil the requirement for a functional paradigm as described above. These are a qualitative model, constructed from natural language, and a quantitative model, comprising mathematical formulae. Now it is the distinction between the two, which causes the disparity over the formal definition of a functional model. The former of the model classes, the qualitative model, is considered to be a true functional model of an artefact. Whilst the latter, the quantitative model, is considered a behavioural or function-behavioural model of the artefact, and may be considered to include additional information which specifies the manner or means by which the required function is achieved. Indeed Johnston and Thornton (1991) detail that a qualitative description is the only way to design a truly original solution, however, the formalisation of such grammar is proving to be difficult for all but the narrowest of domains.

2.2.5 Configuration modelling

Configuration modelling assumes that the set of available components is predefined and the goal is to select components and their interconnection so that all user requirements are satisfied, no design or domain constraints are violated and an optimality criterion is considered (Valasek & Zdrahal, 1997). Therefore, for this class of problem all the components have been designed or selected and the task is how to assemble them into the completed product. Each component is of a known size and each has a certain set of positional constraints. Although these positional constraints may be overcome by the introduction of coupling elements. Not all configuration problems are well defined. For many problems, some of the components to be fitted into the assembly can be altered in size, shape, or function, giving the designer more latitude in determining potential configurations and making the problem solution more difficult. At which stage the introduction of appropriate optimality criterion is essential. A common definition of the configuration task is given by Mittal and Frayman (1989). Although port might be replaced by interface which defines a common point or boundary between two elements.

"Given (A) a fixed, pre-defined set of components, where a component is described by a set of properties, ports for connecting it to other components, constraints at each port that describe the components that can be connected at that port, and other structural constraints; (B) some description of the desired configuration; and (C) possibly some criteria for making optimal selections."

Mittal and Frayman detail three aspects of the configuration task which are:

- One cannot design a new component during the configuration task.

- Each component is restricted in advance to only be able to connect to other components in fixed ways i.e. they cannot be modified to get arbitrary connectivity.
- The solution specifies both the components in the configuration as well as how they are related.

Indeed Brown (1999) continues with a description of the configuration task and in doing so breaks-down configuration into three subtasks:

Configuration = Selection + Association + Evaluation

Here association involves the establishment of the logical relationships between components. Whilst selection comprises choosing components and perhaps some compatibility checking, and evaluation includes complete compatibility analysis and the assessment of goal satisfaction for the configured system.

2.2.6 Feature-based modelling

Ullman (1992) states that for systems, assemblies, or components the term feature refers to specific attributes that are important such as dimensions, material properties, shapes and functional detail. A feature is associated or effected on an object and are therefore unable to exist without a component. Feature based modelling has been introduced to overcome the semantic limitations imposed on designers by CAD systems dealing only with geometry creation and modification. Features are intended as collections of geometric elements and functional characteristics that may be logically grouped together. A typical feature will thus comprise a function(s), its associated variables and governing equations (Johnson & Thornton, 1991). Possible features include: boss, channel, fillet, flange, groove, chamfer, shoulder, thread and web to name but a few (Bordegoni & Cugini, 1997). Several CAD systems successfully integrate feature-based modelling in their systems such as Pro/Engineer (PTC Inc, 2001). Although this tends to be applied to assembly design such as housings and casings once the part geometry and assembly configuration has been achieved

2.2.7 Performance modelling or simulation

Performance modelling and simulation deal with the representation of performance. Where this performance represents a measure of the capability of the considered system to undertake or achieve its desired function. These approaches truncate the conceptual and embodiment phases of the design process and aim to support the designer in rapidly embodying and testing various solutions. This embodiment determines a set of parameters for the system elements which meet the desired performance characteristics for the design. For mechanical systems, this performance may well be measured at a system level, but consideration of performance for assemblies and components must also be undertaken in order to design, select and procure individual components.

Simulation is widely reported (Woolfson & Pert, 1999; Gould & Tobochnik, 1998) often utilizing standard packages for a design through simulation approach. Simulation generally aims to evaluate the performance capabilities of a particular configuration during its operation cycle, and considers aspects such as loads, kinematics and dynamic response. Systems are configured iteratively in order to achieve the desired performance characteristics. In comparison, the work on performance modelling represents outline schemes in terms of individual components, their connectivity or topology and desired performance characteristics. This approach aims to resolve the outline scheme in order to determine a set of fully specified components that fulfil the overall operational and performance requirements for the design. Once the system configuration has been determined, in terms of its arrangement, component types and respective sizings, then the set of components can be used to produce detailed engineering drawings.

2.2.8 Summary

The range of modelling activities and their relative extents of application over the design process is shown in figure 2.7. This highlights performance modelling or simulation as the only approach to provide support for the conceptual, embodiment and detailed phases of design. Configuration modelling and constraint modelling also provide support across the transition from concept to an embodied solution, however, this support is limited. In the case of constraint modelling, the approach, although applicable throughout the various phases of the design of any technical system, is a high level top down one which fails to capture the intricacies of performance and geometry that are necessary for the specification of real[2] components. In contrast to this, configuration modelling provides for the necessary level of detail for both performance and geometry, but at the expense of flexibility. More often than not component types and their associated models are predetermined and the size and layout of the assembly is limited or predefined.

[2] The term 'real' is used in this work to denote those elements which may already exist, have been previously used and have predetermined or predefined properties. In many cases, these are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified.

2.3 Standard components in engineering design

For the purpose of this work, the term 'standard component' is used to represent both standard catalogued components and standard designed components (Culley & Theobald, 1997). Which in today's competitive engineering industry may provide for the majority of components in an assembly. In fact, a recent survey by Theobald (1995) highlighted that standard components can provide for up to 80 percent of the components in a machine system, summarised in figure 2.8. The benefits of standard components are discussed in chapter 1, and include improved quality, reduced cost, known performance capabilities and global acceptance. Furthermore, the desire to standardise may not be purely for competitive reasons. In the aerospace industry approximately 70 percent of designs and their components are adapted or reused because reliability is of paramount importance. However, there are situations where the inclusion of standard components may not be advantageous. These include designs where components are used in very large numbers, technical issues are critical, components are very large or heavily integrated into the structure of the design (Pahl & Beitz, 1984; French, 1988). In these cases it may be more economical to manufacture a new component that satisfies the requirements more precisely. Such components are termed bespoke components (Culley & Theobald, 1997).

The important role of standard components for creating high quality low cost design solutions is discussed in section 1.2. The importance of effectively selecting standard components to deliver the highest quality and performance levels is acknowledged by many researchers and significant research has been undertaken into techniques that improve the searching, identification and selection of a particular component from a manufacturer. The various tasks necessary for the selection of standard components within the overall design process are shown in figure 2.9, taken from Allen *et al* (2000). Developing methods that support these tasks are particularly important because up to 20 to 30 percent of the designer's time is taken up with searching through information to identify a component that fits into the system (Ullman, 1992).

This time consuming and iterative process involves selecting component sizes such that the system performance is attained and more importantly that components are geometrically compatible (i.e. fit together) and matched in terms of their performance capabilities. The ability to undertake this system evaluation is severely frustrated by the fact that standard components follow a discrete, finite range. This demands that the designer arbitrate various component sizes to ensure that critical parameters between components are matched, such as the diameter of a shaft and the internal diameter of a bearing. To achieve this, the designer must manipulate much of the selection data manually and enter it into the appropriate electronic representation for the design and selection of the desired component. To determine a compatible solution which meets the PDS the designer may have to undertake many iterations. Furthermore, changes in a single component later in the process can affect many other components. This may demand that the designer return to the iterative selection process to determine a compatible system of components. A systems approach is shown in figure 2.10. The traditional or manual process demands that the designer manage all the selection data. This involves interrogating component representations, arbitrating conflicting parameters and assimilating and manipulating component selection data[3], all of which occupies a large proportion of the designer's time.

Ullman (1992) and BS 7000 (1999) are alone in identifying the activity of component selection as a separate, distinct and important activity. Ullman proposes that catalogue selection systems will have a significant effect on the way designers organise their design processes, and that the use of catalogues provides both a source of possible design solutions as well as the descriptive information a designer seeks for a specified component. In fact, a review by Boston *et al* (1999) indicates that supplier literature is the foremost source of information for designers. Because of these factors, the activity of component selection and the modelling of components for selection are fast becoming one of the <u>most important</u> activities in systems design and modelling respectively. In fact, authors such as Culley & Vogwell (1990) have developed systems for the automated selection of components including coil springs, seals and bearings. Their research culminated with the development of the CASOC system for the selection of bearings, covering 22000 sizes from 10 manufacturers (Vogwell & Culley, 1991). Selection design is defined as selecting or choosing one item or more from a list of similar items. Such activities require catalogue searching where there are many items with many different features. The potential number of solutions generated can be vast and each must be evaluated against the specified requirements. To address this, manufacturers are implementing electronic catalogues, the benefits of which include:

- Efficient storage and distribution of information with CD-ROM based catalogues and Internet-hybrid systems.
- Fast and efficient retrieval of component information and associated data.
- The ability to automate analysis procedures and sizing calculations.

[3] Selection data encompasses the range of attributes upon which a particular standard component is specified.

- The ability to provide an interactive environment with best practice suggestions for coupled components and fitting etc.
- The inclusion of two-dimensional and three-dimensional drawings primed for the exportation into CAD environments.

Many catalogues now include basic heuristics and integrated selection processes in order to determine the best component match. There are however, some limitations to electronic catalogues and their associated *modus operandi*, discussed by Wallace (1995). These include variations in data formats between catalogues, irregularly ordered data, varying selection strategies and the manner in which subjective factors are handled. Also, the fact that many tools which deal with selection design consider components in isolation demands that the designer still manually manipulates and arbitrates selection data in order to ensure a compatible set of components is determined. As yet, there are no computer based methods or tools available which provide for the support or automation of a systems approach for selecting standard components from electronic representations. Although there are some tools for the support of conceptual and embodiment design. These are discussed in section 2.4.

2.4 Computers in the design process

The importance and dependency on computers for today's engineers is highlighted by Whitney (1990) who proposes that computers are the main advance and driving force of design today. Computers in engineering have many and varied applications spanning the overall product life cycle. These include Computer Numerically Controlled (CNC) machining (Smith, 1983), product modelling, rapid prototyping (McMahon & Browne, 1993), multimedia catalogues (Culley & Webber, 1992), Computer Aided Process Planning (CAPP) (Vonderembse & White, 1991) as well as numerous analysis procedures, such as Finite Element Analysis (FEA) (Moaveni, 1999) and Computational Fluid Dynamics (CFD) (Anderson, 1995). As computational techniques, hardware and software capabilities continue to advance more detailed modelling and analysis procedures for engineering design are developed.

Computational tools afford a tremendous saving through the removal of routinisation and the reliable treatment of mathematical relationships, thus freeing the designer's time to extend and refine concepts. It is plausible for computational tools to be effected at any stage of the design process following the clarification of the task. At different stages of the design process varying levels of modelling may be introduced which aid the designer in the search for a solution(s). Earlier work by Pahl and Beitz (1984) separated the creative aspect of design from the computer and suggested an interactive or modular system between designer and computer. This separation occurs predominantly in the conceptual phase of the design process and the proposed scheme implements the computer for primarily storage purposes only. Since these early years, the CAD market has rapidly evolved, driven by the ever increasing demands of the engineering industry. The scope of some CAD tools over the design process, both commercial and research based, is shown in figure 2.12.

Computer Aided Design (CAD) is generally perceived to be a subset of the design process in the typical product life cycle, figure 2.1, whilst its close relative Computer Aided Manufacturing (CAM) is a subset of the manufacturing process. Zeid (1991) defines CAD tools as the intersection of three sets; geometric modelling, computer graphics and design tools. This definition is perhaps somewhat limited and may now be considered to encompass modelling *per se*, computational methods and design tools, illustrated in figure 2.11. The set definitions implemented here are broad enough so as to encompass many of the specific methods or technologies that the reader may be familiar with. The basis of CAD tools is therefore the augmentation of design tools (analysis codes, heuristics procedures, design practices etc), modelling techniques or engineering representations within a computer based environment in order to achieve the design goal reliably and efficiently.

The extents or scope of usefulness for the majority of commercial tools considered in this work focuses on the detailed design phases of the process. At which stage the concept, its included elements and the topology or arrangement has been determined. These tools are commonplace in most design offices, and support the specification of surface finishes, parts listing, tolerances, scaling and automatic dimensioning as standard. Most packages provide the user with two-dimensional layout or drafting tools and the facility to construct three-dimensional models. As a consequence, there are not many commercial tools which deal with the <u>concept</u> and <u>embodiment</u> phases of the process. Of the support tools reviewed, 'design through simulation'[4] tools such as AMESIM (AMESim, 2000) and SPICE (Keown, 1994), are the only approaches that span the concept to embodiment and detailed design phases. However, these tools are specifically for fluid power systems and electrical circuit design respectively, and are not applicable to machine systems and standard catalogued components.

[4] Design through simulation is used to describe the approach of iteratively changing component specifications to achieve the desired system performance.

2.4.1 Computer based support during the early stages of design

Computer based support for the early stages of the design process is dominated by research tools. The majority of these tend to focus on the tasks associated with the conceptual phase of the process, and fail to fully bridge the gap between the conceptual and embodiment phases.

To support conceptual design, Brady and Juster (1997); Mantyla and Gui (1994) and Johnson and Thornton (1991) describe a range of computer based tools. Brady's work proposes a suite of programs known as the Conceptual Assembly Design Tool (CAD_T) which allows the designer to create outline geometry from function structures, thus providing a bridge between conceptual and embodiment design. The user specifies a function structure which must then be outputted as a search request to the Assembly Interface Entity (AIE) Library. Each AIE comprises three elements:

1. A function name relating it to a particular assembly interface function.
2. A solution principle such as a seal or flange.
3. The final element is a parameterised instance of three-dimensional solid geometry for the solution principle.

For each function specified by the user there may be one or any number of solution principles suggested by the system. Indeed, future work by Brady and Juster aim to enlarge the AIE library as well as enabling user defined AIE's. By using inheritance in the AIE library it is possible to create generic objects, the example used is for a flange and a spigot which combine the fixing and locating functions of each to create a spigoted flange. AIE's are linked by the user and the current Assembly Interface Entity Relationship is depicted graphically in the graph editor. Once a solution principle has been created a solid model of the conceptual assembly may be formed.

Mantyla and Gui (1994) also propose a support tool for the early phases of design, again considering functions and features. Their aim is to bridge the gap between feature and function. The system aims to support both conceptual design at high levels of abstraction and feature modelling at low levels. Mantyla and Gui again uses functional decomposition combined with a classification of components with respect to their performance of desired function in order to build a design solution in terms of a function-orientated view. Bond graph techniques may be used to analyse forces (Gawthrop & Smith, 1996) and a module-orientated view can be achieved when the hierarchical functional-view is converted into a physical structure of the device expressed in terms of subassemblies, parts and features. However, this conversion has not to date been attained by Mantyla and Gui.

Ward and Seering (1989) present a mechanical design 'compiler' to aid the design engineer in composing new designs and then producing an optimal set selection of components suitable for use in the detailed design stage. Firstly, the new design is inputted into the software environment as a high-level description, a specification and a cost function. This design is then compiled or transformed from a schematic representation into an optimal set of abstracted catalogued components. The abstracted catalogued components are grouped according to catalogue number (series) into a hierarchical structure. These groups are known as sets and are all subsets of their master set, for example in the domain of electric motors the master set would be motors with subsets; ac and dc, these subsets of motors would also possess their own subsets. The 'compiler' firstly eliminates incompatible catalogue components, following which the user defined specifications and governing equations of components are propagated and executed to eliminate subsets of the compatible components, leaving only a small number of catalogued components and combinations that meet the requirements at the least cost.

One commercial computer based tool which is presented as the environment for innovative and creative design is IMPhenomena (Davis, 1998). Although not providing analysis and design of systems it does provide the user with a vast range of functional and physical effects from its included database. The system classifies technological effects according to the objective sought. This provides the user with a very broad interdisciplinary perspective of possible solutions to a design problem. Concepts can be produced very quickly with accompanying examples of individual elements and also suggestions of new and often extraordinary connections (Davis, 1998).

The Schemebuilder system, developed at Lancaster (Schemebuider, 1998), comprises a suite of software tools which enables the user to create a range of conceptual design solutions. The Schemebuilder system presently covers the hydraulic and pneumatic domains, the user enters both qualitative and quantitative requirements in order to generate a number of system schemes which follow best practice knowledge and design methodologies (Schemebuilder, 1998). The designer can then evaluate the schemes (aided by a qualitative ranking system) and select the most promising which represents a real solution down to a component level. The user defines the qualitative attributes by selecting from a list the descriptions which best define the desired load attributes of the system. A range of conceptual solutions is then proposed by the system, following which the user can choose to size elements of the system by entering quantitative data, such as load attributes, stroke, maximum speed, maximum static force and load mass.

During the embodiment phase of machine systems design and in particular transmission systems, work by Culley & Theobald (1997) aims to provide support for the designer. The approach allows the user to incorporate standard designed or standard selected elements into an assembly model. Where this model represents the system performance and coupling conditions between system elements. The approach incorporates parametric sizing and selection algorithms for individual components, some analysis procedures and limited optimisation routines. This approach impacts on this work and is reviewed in section 3.3.1. Methods which deal with system embodiment are particularly important, as these tasks can demand a much greater proportion of the designer's time than tasks in other phases of the design process (Pahl & Beitz, 1984). Whilst some tools may provide support for the designer in embodying a solution concept they do so in a very restricted manner. The approach adopted by Culley & Theobald (1997) is to specify desired elemental characteristics or parameters, the software then resolves the system in order to determine a set of component sizes which meet the desired characteristics. However, the arrangement of components and types is severely restricted and only a single assembly of limited size may be considered.

2.4.2 Summary

The computer based tools discussed in this section address many different tasks of the design process. The majority of commercial tools support the latter stages of the design process. Consequently, support over the transition from conceptual to embodiment stages of the design process is limited. Of the tools which support this transition, many are research based and are severely restricted in their capabilities or are not applicable to the mechanical domain, and in particular machine systems design. Furthermore, the important area of systems design with standard components is somewhat overlooked for all but the selection of individual components in isolation. A number of modelling approaches to support the transformation of a concept to an embodied solution are reviewed in chapter 3.

2.5 Identification of research issues

This chapter has provided an overview of research in engineering design and a number of key mechanisms for improving the efficiency and effectiveness of engineering design, including design process models, modelling and analysis approaches, computer based support tools and the utilisation of standard components. These are not exhaustive but are identified as the major areas that directly impact on this body of research.

The importance of standard components for the commercial success of the product, the manufacturer and the end user are described, and the need to provide support for standard components during the early stages of design is discussed. These early stages of the design process comprise the important phases of conceptual design and embodiment design as discussed in section 2.1.

The discussion of design process models and activities highlights the fact that much of the designer's work involves the adaptation or variation of existing assemblies to meet a new, or changed set of requirements. For such activities the desired changes in performance can often be attained by incorporating various standard component types and/or sizes. Therefore, the ability to consider changes in the performance of assemblies or systems due to changes in the specification of individual components is very desirable. Current technologies and approaches for the undertaking of such tasks demand that the designer performs much of the computationally intensive and often iterative procedures manually.

The review of various modelling approaches for engineering design, highlights the fact that few of the approaches provide for support during the early stages of design and in particular, the transformation of a concept to a fully embodied solution. Performance modelling or simulation are the only approaches which consider the necessary characteristics of the system and attributes of individual components for complete embodiment. For this work, a systems approach is necessary to determine an optimum solution, and consideration of individual components is essential for representing the nonholomonic nature of standard components and their electronic representations.

Furthermore, the consideration of both performance and geometry is essential for the effective sizing and selection of standard components.

- *Performance* considerations ensure that energy interfaces are compatible and the magnitudes of energy transfer are acceptable output and input levels for coupled components.
- *Geometric* considerations ensure that geometric interfaces are matched, so that components can fit together and energy can be transferred across the interface.

Simulation techniques adopted in the fluid power and electrical industries do provide for a systems approach, however, the relationship between geometry or topology and performance is not as complex as in the mechanical domain. These simulation approaches are discussed in some detail in chapter 3. The need to consider both performance and geometry at a system level combined with the fact that standard components are generally only available over a finite, discrete range of sizes frustrates a systems modelling approach. It is postulated that these factors are the reasons why few modelling approaches and CAD tools, support the transformation of conceptual systems to fully embodied solutions.

The few research systems that do claim such features often perform them in a very prescribed manner, which restricts them to purely configuration tasks. That is to say a design solution can only be represented by a predefined range of elements and in a prescribed manner, these factors mean that such tools are not suitable for the conceptual phase of the process. Other approaches use continuous parametric models to represent standard components, which means that the designer is not dealing with 'real' components. Consequently, the designer will have to deal with the various representations for third party components later in the process. Which because of the discrete nature of standard components may demand significant redesigns or reconfiguration of systems later in the process. In addition to this, section 1.2 discusses the advantages of electronic component representations (catalogues) for selecting a particular component, it is therefore desirable to consider these technologies within an overall systems approach.

The limitations of current approaches and the outstanding research issues identified in this chapter provide a set of requirements for a new system modelling approach for machine systems.

- The ability to represent a mechanical system of any size and configuration of components.
- An approach that can represent the performance of individual components and evaluate the performance capabilities of the system in its entirety.
- An approach that provides for the consideration and selection of standard components.
- The ability to analyse and compare concept variants.
- A methodology that does not prescribe a modelling approach for component representations *per se,* but provides for the inclusion of new and emerging electronic representations.
- A computer based support tool which provides a platform for the optimisation of component mixes.

To address these important outstanding research issues an *integrated modelling approach* is proposed. An integrated modelling approach builds on a number of modelling techniques reviewed in section 2.2. An integrated modelling approach is defined in this work, as an approach which represents the performance of a system by manipulating and integrating the various representations for individual elements.

From the issues identified above three hypotheses are proposed which are addressed by this work.

Hypothesis 1

The electronic representations for standard engineering components can be manipulated in such a manner so as to enable the performance of mechanical systems to be represented.

Hypothesis 2

This approach can be implemented in a computer based support tool to enable the representation of topology and performance for conceptual systems of standard components.

Hypothesis 3

The approach can be extended to enable the configuration, embodiment and optimisation of engineering systems from standard components.

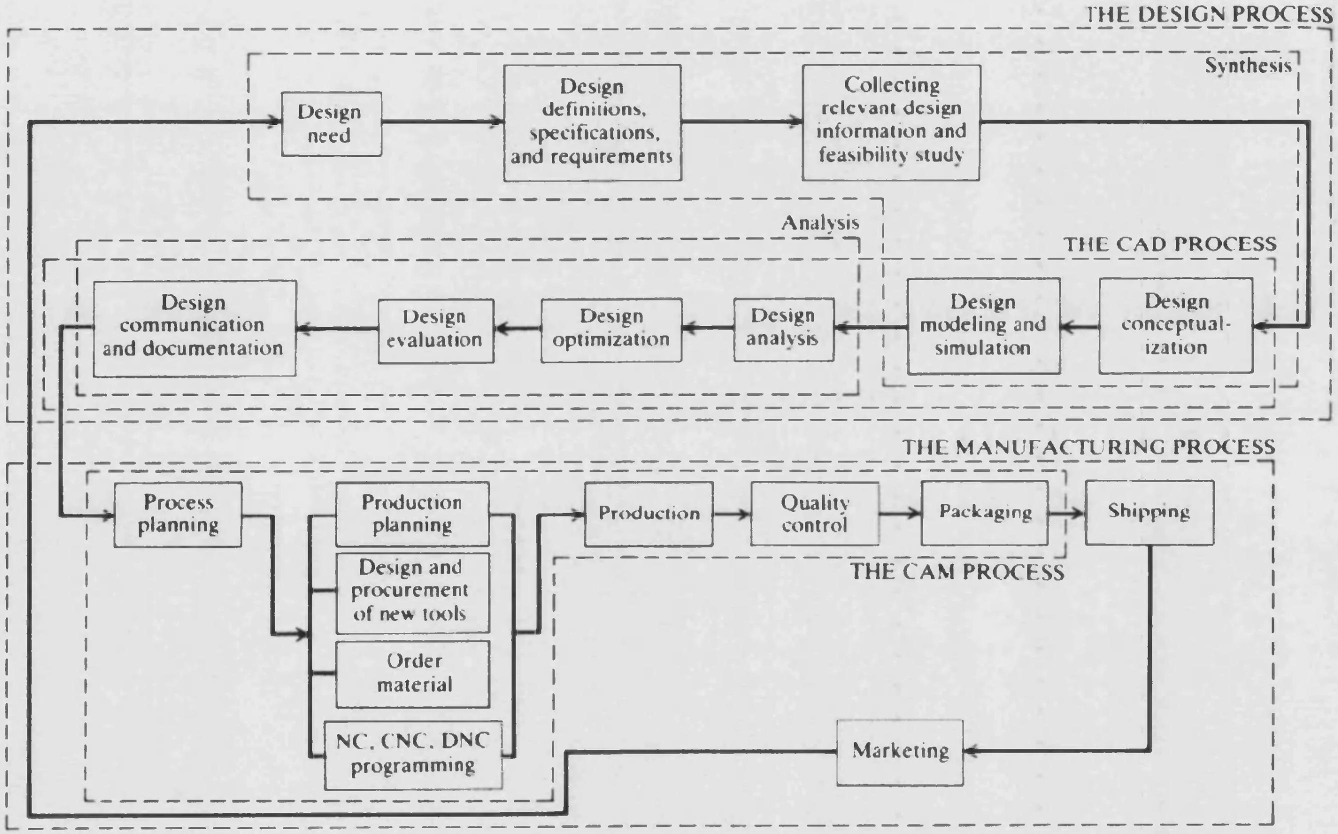


Figure 2.1 – The typical product life cycle (Zeid, 1991)

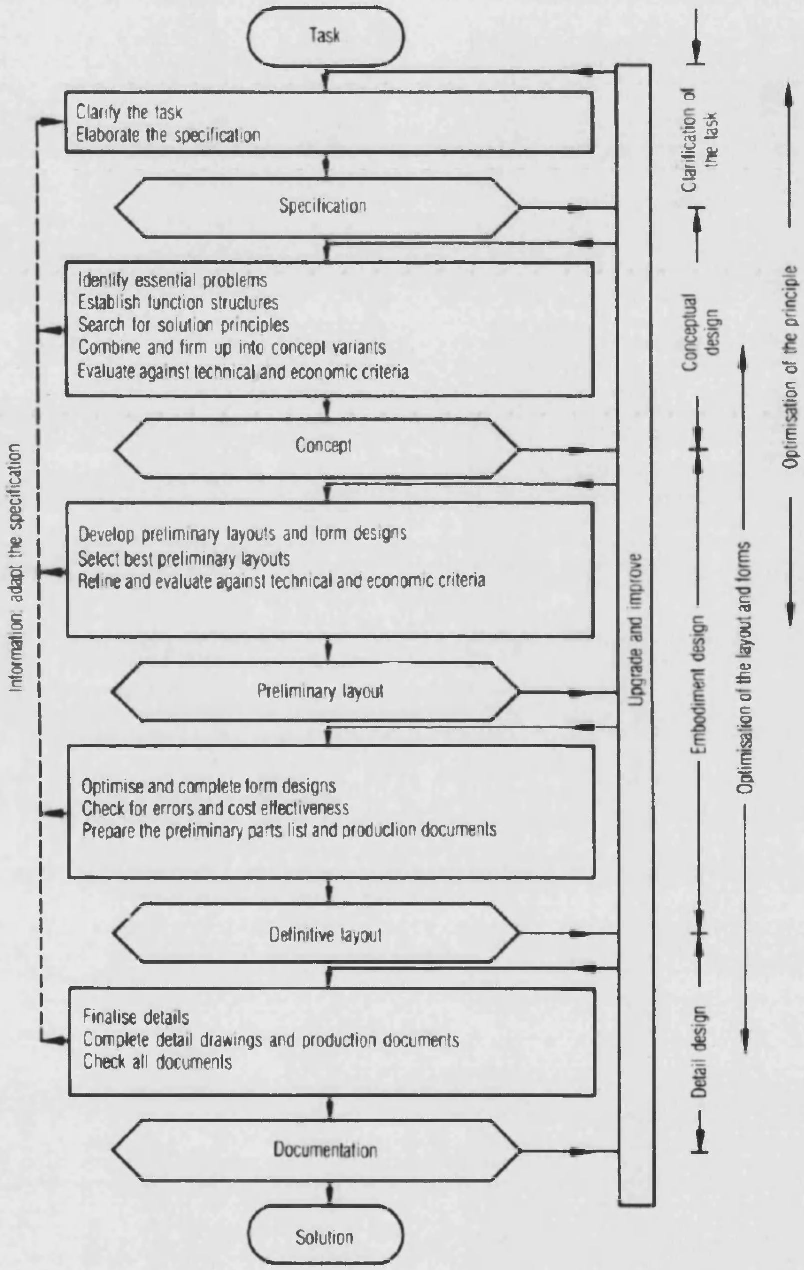


Figure 2.2 – Design process model (Pahl & Beitz , 1984)

Specification Development/Planning Phase

- Understanding the design problem:
  - Developing customer requirements
  - Assessing the competition
  - Generating engineering requirements
  - Establishing engineering targets
- Planning for design

Conceptual Design Phase

- Generating concepts:
  - Functional decomposition
  - Generating concepts from functions
- Evaluating concepts:
  - Judging feasibility
  - Assessing technology readiness
  - Go/no-go screening
  - Using the decision matrix

Product Design Phase

- Generating the product:
  - Transforming existing products
  - Embodying the functions
  - Designing product and production concurrently
  - Patching and refining the product
- Evaluating the product:
  - Monitoring functional changes
  - Evaluating performance:
    - Using experimental models
    - Using analytical models
    - Optimizing Design Group
    - Using robust Design Group
- Evaluating costs
- Designing for assembly
- Designing for other "itilities"
- Finalizing the product

Figure 2.3 – The design process an organisation of techniques (Ullman, 1992)

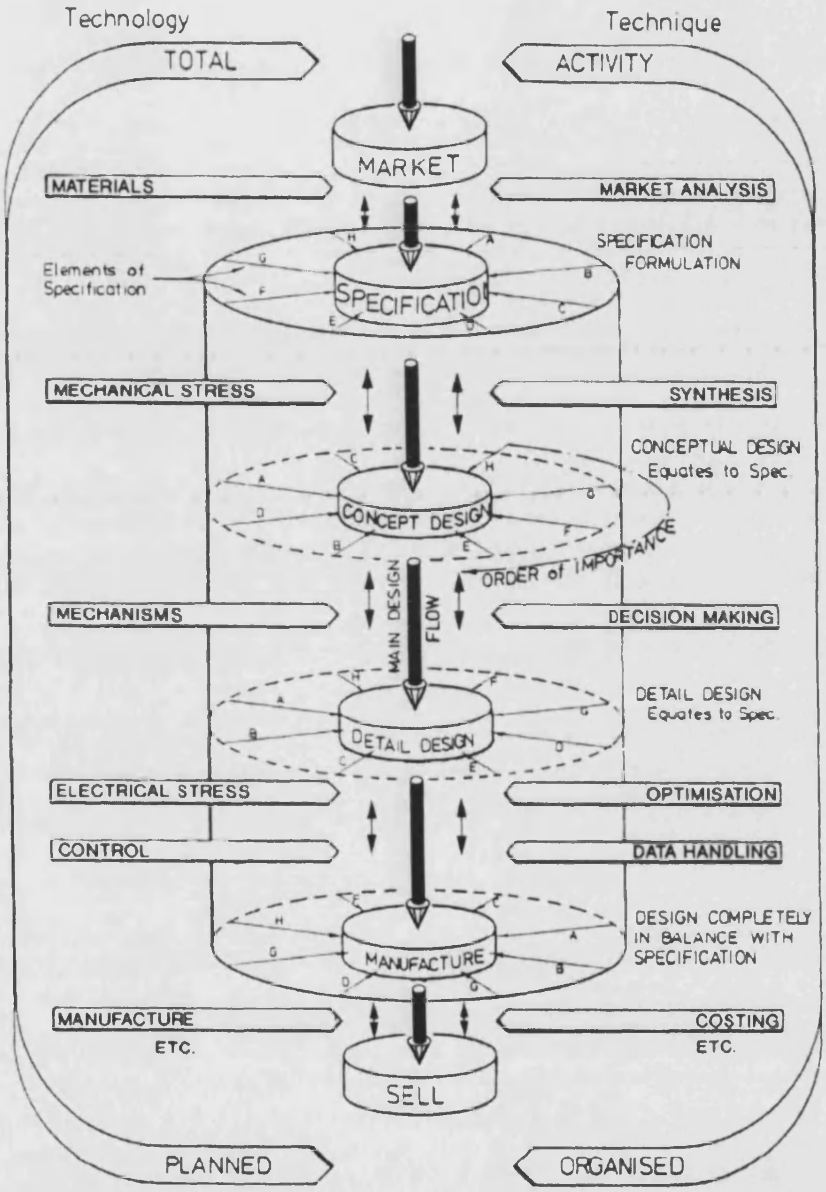


Figure 2.4 – Design process activity model (Pugh, 1991)

Type of Design Activity	Definition	Author/source							
		Pahl & Beitz	Ullman	Pugh	French	Gero	BS 7000	VDI 2221	Ulrich & Eppinger
Original	Elaborating an original solution principle for a system with the same, a similar or new task	●	●					●	
Adaptive	Adapting a known system to a changed task	●						●	●
Creative	An original design solution to an existing or new problem		●			●			
Variant / customized	Varying the size and/or arrangement of certain aspects of a chosen system	●							●
Fixed Principle	Solution principle and design are the same dimensions of individual parts are changed	●							
Redesign/ Development	The modification of an existing product for a new set of requirements		●			●		●	
Routine	Design of the artefact can be represented by a system/network of rules and equations		●						
Non-routine	Not all of the design and structure variables, performance and behaviour variables are known at the outset					●			
Mature	Complete knowledge about the design problem exists and design focuses on asthetics and optimistaion		●						
Static Product	Design changes are incremental or non-existent			●			●		
Dynamic Product	Design changes are innovative and frequent			●			●		
Evolutionary	Continuous product improvement to meet slowly changing needs or evolving science and technology.			●	●		●		
Non-evolutionary	Deliberately innovative design using new technology				●				
Catalogue	Selecting and assembling of catalogue items						●		

Figure 2.5 – Classification and definition of design activities

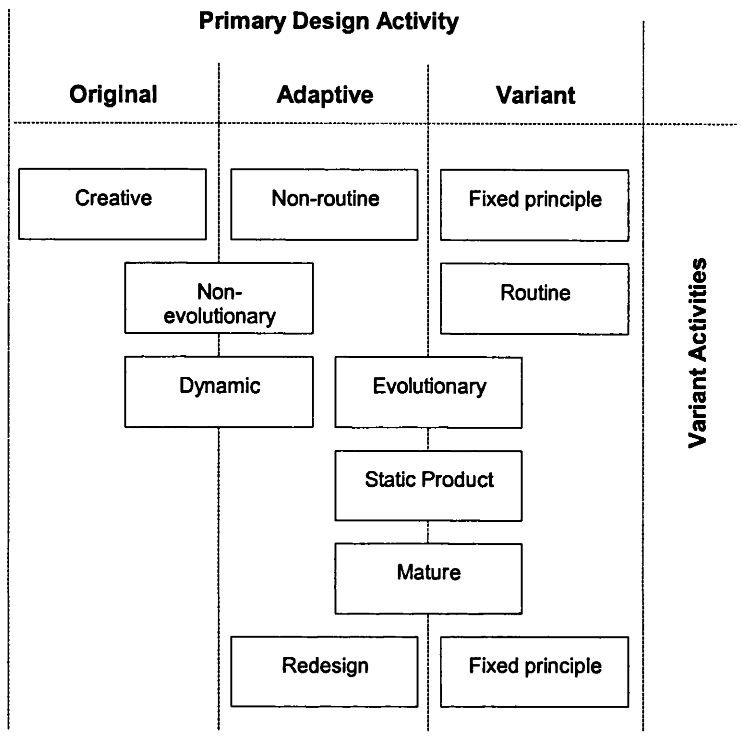


Figure 2.6 – Primary design activities and their combinatorial variants

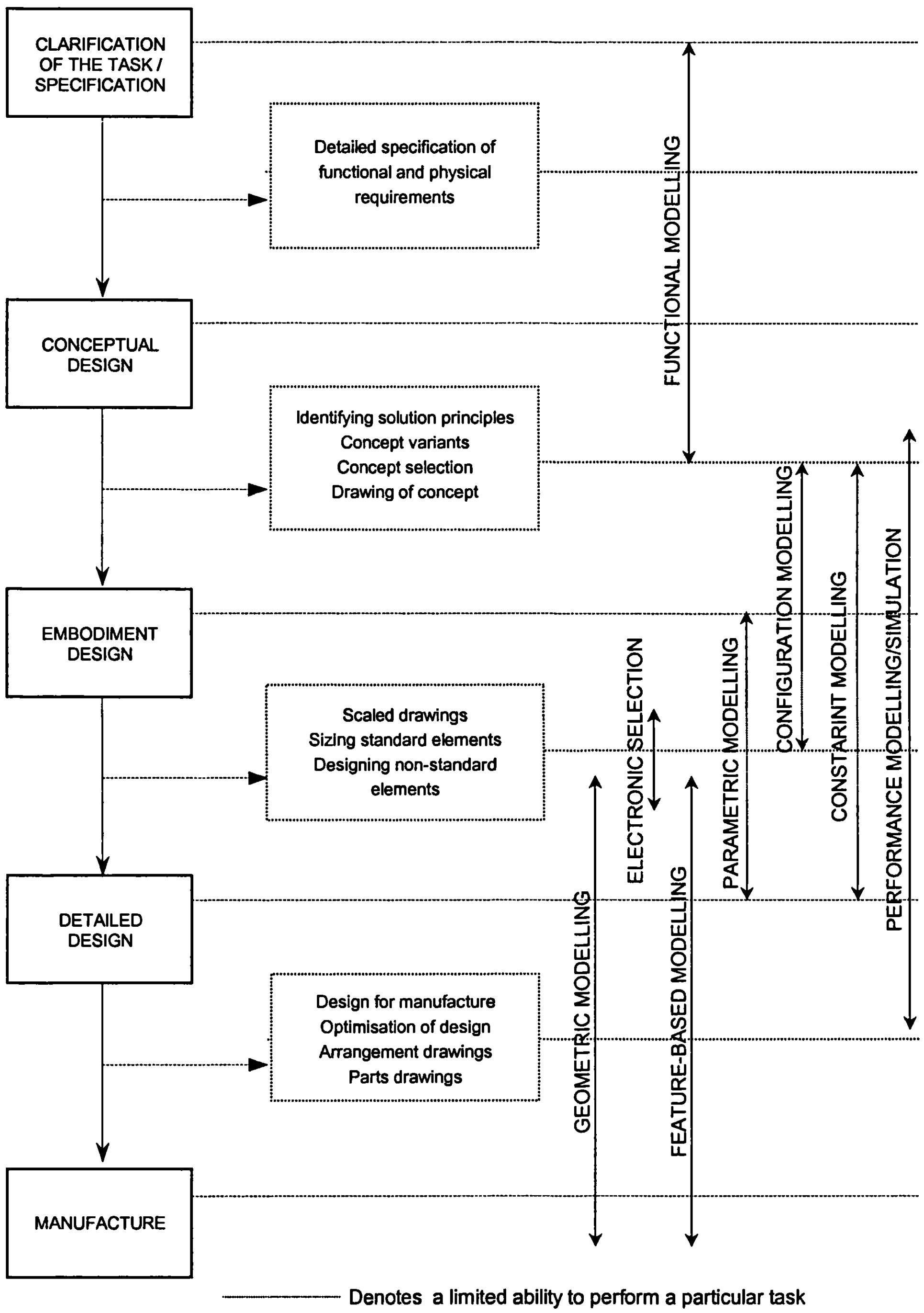


Figure 2.7 – The scope of modelling and analysis activities over the design process

	Machine system		
	Hydraulic Press	Rotapactor Rock Crusher	Hamworthy Oil Burner
No. of components	172	32	185
No. of standard components	96	18	151
Percent standard components	55.8 %	56.5 %	81.6 %

Figure 2.8 – Utilisation of standard components in machine systems

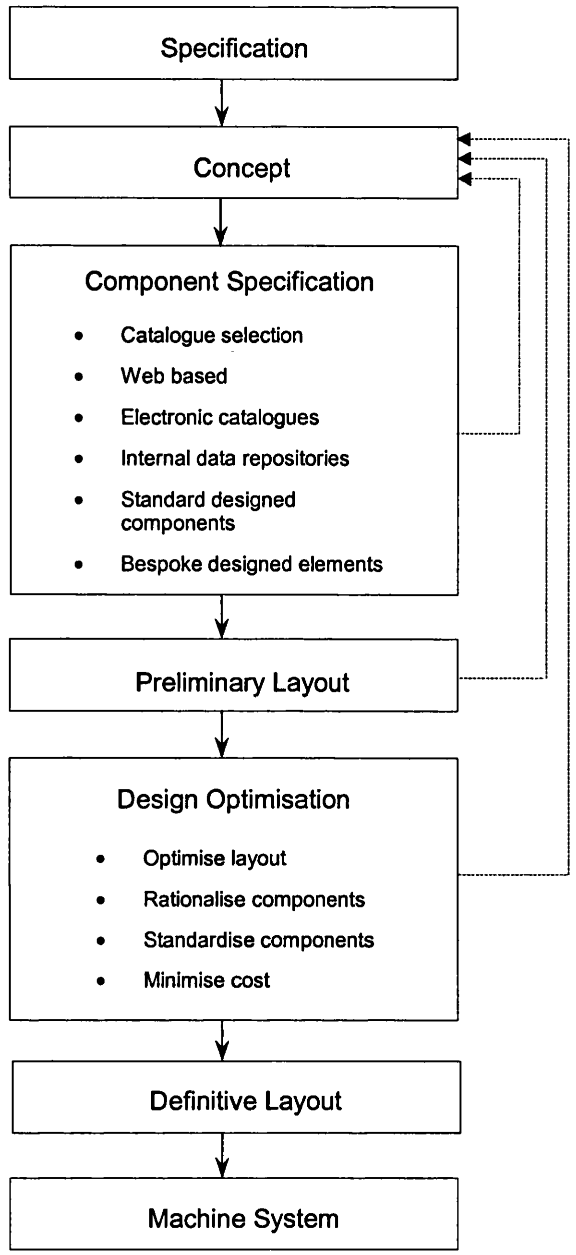


Figure 2.9 – Component selection activities within the design process (Allen *et al*, 2000)

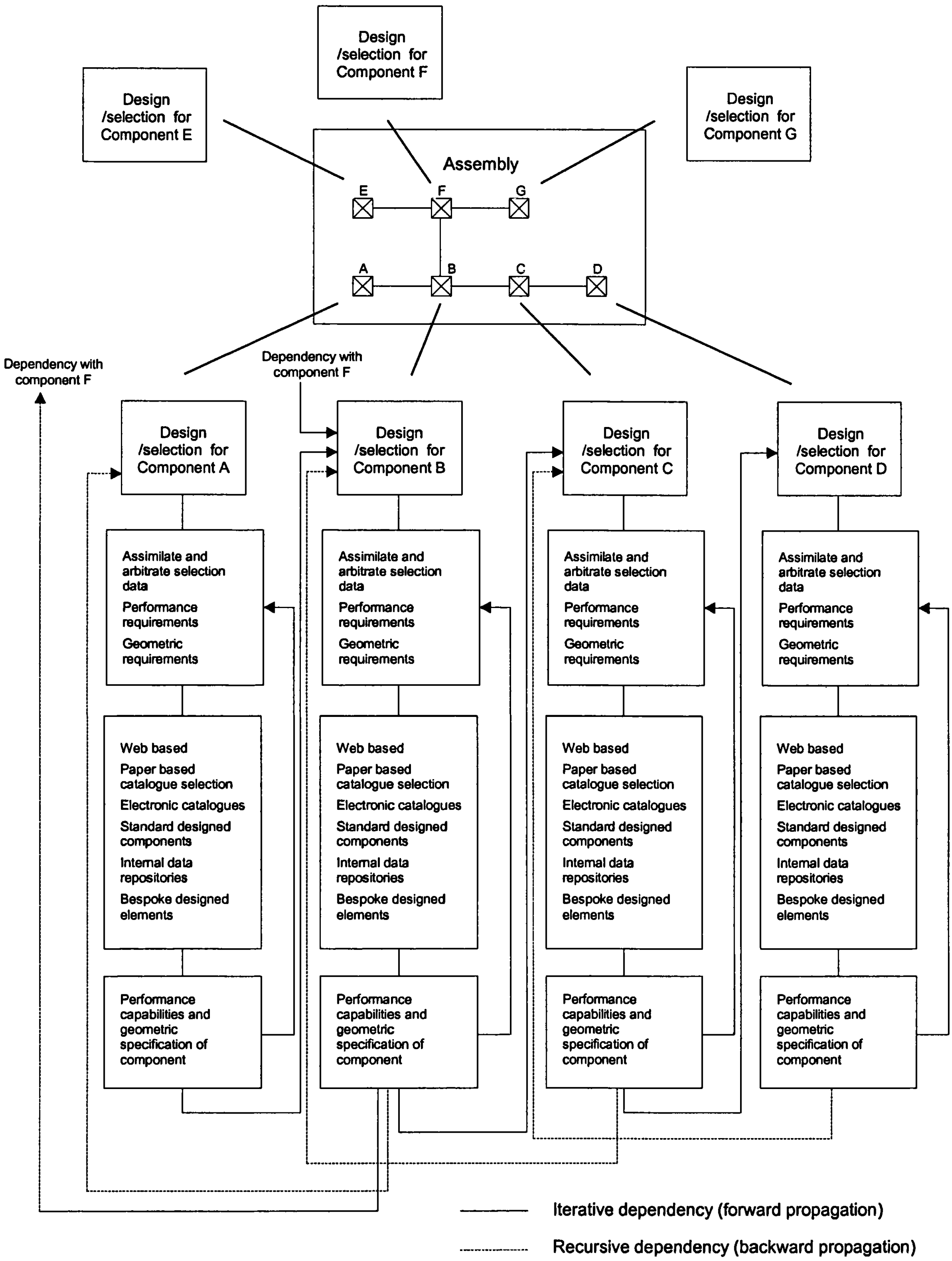
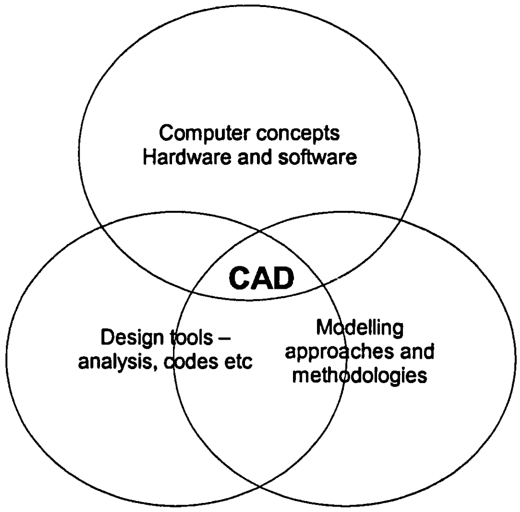


Figure 2.10 – A systems approach to component selection for part of an arbitrary assembly



Computer Aided Design (CAD) tools are bounded by the region which defines the intersection between modelling, design tools and computational concepts.

Figure 2.11 – The definition of CAD tools based on their constituents

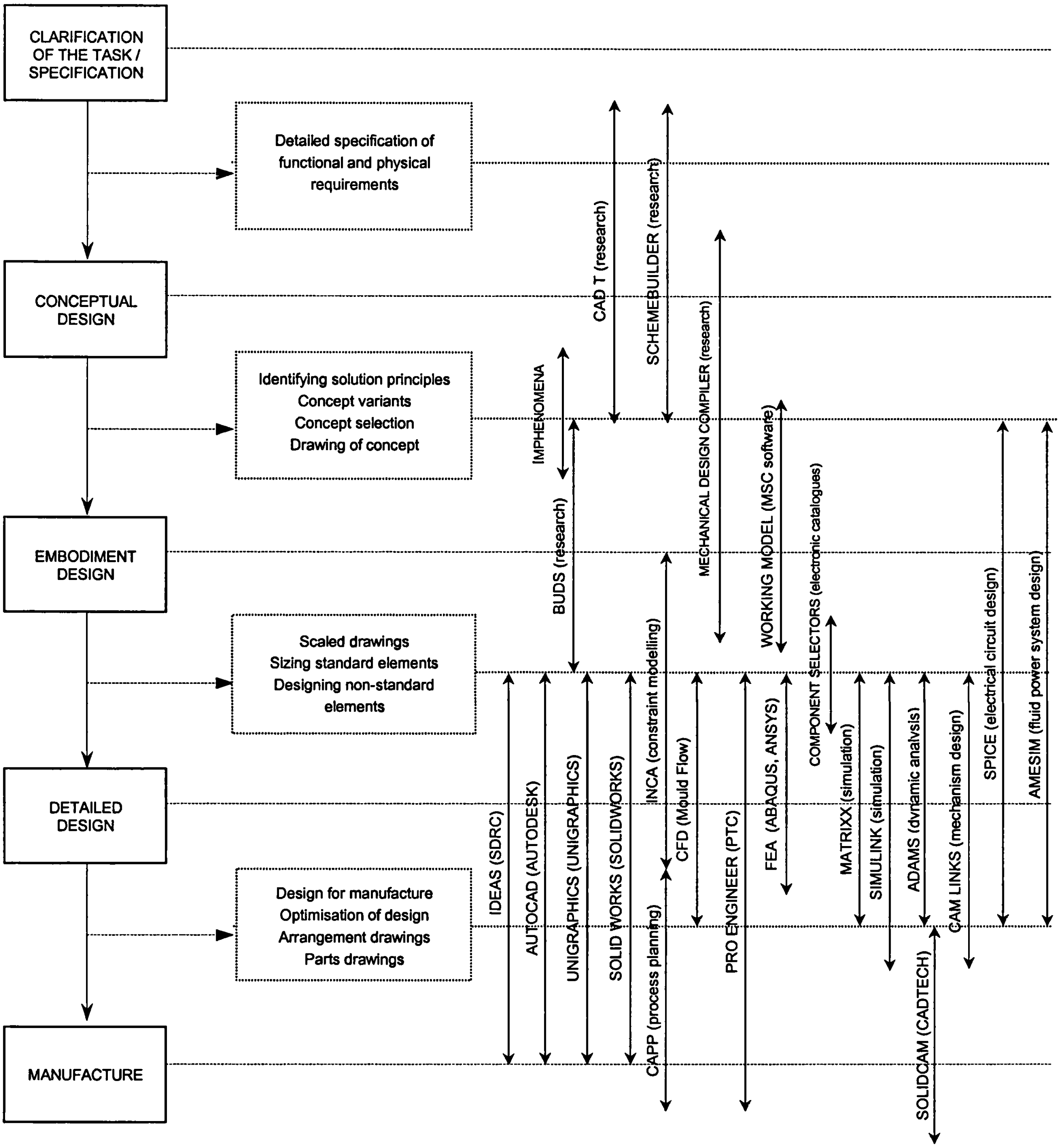


Figure 2.12 – The scope of some Computer Aided Design tools over the design process

Chapter 3

Modelling and analysis of engineering systems

This chapter appraises current modelling approaches for representing engineering systems and discusses the various standards for representing engineering components. This appraisal discusses the capability of these existing technologies and approaches to represent machine system concepts for embodiment with standard components. Three distinct areas or technologies are identified which impact on this work; standards or methods for representing engineering components and systems, methods for the electronic representation of standard engineering components, and system modelling approaches developed for the performance modelling or simulation of engineering systems in other engineering domains.

3.1 Representing engineering components and systems

Modelling techniques in engineering design are many and varied, some authors distinguish between data models, information models, process models and mathematical models (Court, 1995). Ultimately the modelling techniques themselves may have been tailored to represent a particular aspect, such as information or processes (IDEF methods KBES Inc, 2001). However, modelling approaches all possess one common objective, that is to represent the individual elements, the dependencies and the relationships which constitute the considered system. The models generated are simplified descriptions of a system to assist in calculations or predictions, and the goal of the modelling will drive the format and content of the representation. The techniques reviewed in this work are bond graphs and the STEP protocol, and although fundamentally different these methods are two of the most widely used technologies for representing engineering systems. These technologies are evaluated with respect to their capabilities for modelling essential performance and geometric data necessary to select and fully specify an engineering component.

3.1.1 Bond graphs

Bond graphs attempt to provide a general approach to modelling and system representation that unifies physical systems of all energy domains in the field of engineering, from thermodynamics to electrical circuitry and fluid power systems (Gawthrop & Smith, 1996). Conventionally bond graphs are constructed to represent energy transfer and translation between components. Hence, this approach shifts the attention away from the element which manipulates energy and towards its interaction within the system, and in particular the energy interaction. The energy bond carries all the information necessary to describe this interaction and includes four ideal elements which describe the energy manipulation of any physical component.

The approach of the bond graph is to analyse the system in terms of its constituent parts within a defined boundary system (Gawthrop & Smith, 1996). That is to say the system to be considered is abstracted to a system or set of standard elements that are connected in a structure appropriate to the system process. A popular method of bond graph structuring is to construct an electrical analogue of the system process (Thoma, 1975), and this approach is still widely used today.

Bond graphs which implement energy as the transfer variable require the extraction of two co-variables in each energy domain, defined as ***effort*** and ***flow***, (***e***) and (***f***) respectively. Yielding

$$\text{Energy, } E = \int ef \, dt \qquad (1.1)$$

In modelling with bond graphs voltage, pressure and force are considered to be effort and their corresponding flows are current, flow rate and velocity respectively. The energy (effort and flow) is exchanged through so-called ports on each ideal element and each port represents a single distinct energy interface. These ideal elements in the energy model are categorised into four basic types according to their physical laws and constitutive relation. The first of these elements, the energy transfer element, determines the structure of the model and the flow path(s) through the system for the transfer variable. In contrast, the other three elements; sources, stores and dissipaters are the elements which afford the building blocks for the system process and essentially emulate the system behaviour. The constitutive properties of the elements are expressed as algebraic equations, linear or non-linear, which relate the effort to the integrated flow or vice versa. These element types are summarised in figure 3.1.

In the construction of bond graphs there are two main bond types implemented: the energy bond denoted by a half arrow indicating the direction of energy flow, and an activated bond (signal) represented by a full arrow which has the characteristic that co-variables are conveyed not transferred, figure 3.2. Bond graph junctions can be one of two types, effort or flow, each capable of containing one of the four structural elements detailed in figure 3.1.

Once constructed, a bond graph may be analysed systematically by the introduction of a set of causality rules. Causality rules are in themselves derived from the fact that generally system inputs should cause system outputs. If these causality rules enable the system variables to be explicitly computed then the system is said to be causal. If on the other hand no solution is possible then the system is over-causal and if the system variables cannot be explicitly computed but require the solution of simultaneous equations then the system is under-causal. The assignment of causality to a bond implies that either the effort or flow on that bond is known and thus may be propagated through the graph. In addition to this, the only elements which can force causality are the effort and flow sources and structural elements. The assignment of causality and the execution of constitutive relations for the structural elements permit the system dynamics to be fully represented and a mathematical model to be established.

In order to represent mechanical systems and in particular machine systems, bond graphs must represent certain key performance and geometric parameters to enable the specification of a set of standard engineering components. The use of energy bond graphs permits systems covering several domains to be modelled in a consistent manner and include non-linear and time-dependent behaviours, however, the focus for modelling machine systems and their assemblies tends to be on components from either the linear, rotary or combined motion domains (Culley & Theobald, 1997). This demands that forces, torques, translational and rotational velocities be considered.

Mechanical elements in the energy bond graph can be modelled by considering imposed forces and torques as 'effort sources' and imposed linear or angular velocities as 'flow sources'. Translation and rotational mechanics are dealt with together as so much of the terminology is common. The distinction between kinetic and potential energy sources is achieved by considering springs as flow stores 'C' and masses as effort stores 'I'. Friction may also be included in the model as an energy dissipator and denoted by 'R', see figure 3.3 for an example of such mechanical elements. The implementation of such a scheme is used by Blundell (1982) to model a disc brake assembly shown in figure 3.4, as well as a range of mass, spring and lever systems.

The suitability of bond graphs for system design has been discussed elsewhere; Brown (1991), Karnopp (1982) and Redfield (1996). Bond graphs provide a very powerful tool for modelling of physical dynamic systems and also contain many variables of interest to the engineer such as forces, velocities, displacement, energy and power which are all required for system design. However, Redfield highlights the two main drawbacks of bond graphs in mechanical design. These are the potential size of the graph as systems increase scope and the abundance of derivative causality that hampers equation formulation for geometrically constrained mechanical systems. To mitigate the first of the problems vector bond graphs have been used successfully by Breeveld (1982). The second and most restrictive of the drawbacks is approached by Redfield

(1996) who incorporates a Lagrangian development with bond graphs based on energy and generalised coordinates. Redfield concludes that in real engineering more than the motion variables are required, such as bearing and contact forces which are not directly available. This is certainly the case where configuration and ultimately component sizing information is required from the system representation, and is probably one of the main reasons why such techniques are not extensively used for machine systems design and the design and selection standard components.

3.1.2 Standards for product representation in engineering design

The need to exchange data that describes engineering components is directly related to the need to integrate and automate the various activities involved in the design and manufacture of a product. This exchange is necessary because of the many specialised tools and individuals involved in the various design and manufacture activities. To address this issue, a number of standard formats for representing products have been developed, and continue to be developed in order to include data necessary to describe the complete product. Current standards typically provide for four basic sets of modelling data; design, shape, non-shape and manufacturing. These sets are defined by Zeid (1991) as:

- Design data encompasses the analysis data such as mass property and finite element meshes.
- Shape data consists of geometric and topological information as well as form features.
- Non-shape data includes graphics data, global data such as measuring units and resolution.
- Manufacturing data comprises elements such as tool paths, tolerancing and process planning.

One of the first standards to emerge was the Initial Graphics Exchange Specification (IGES), which dealt primarily with the shape and non-shape classes of data. Further developments incorporated complete product descriptions such as the Product Data Exchange Standard (PDES) that deals with CAD-to-CAM as well as CAD-to-CAD exchange. The most notable of these today is the Standard Exchange of Product data (STEP). STEP is an International Standard for the computer-interpretable representation and exchange of product data (ISO10303) (Pierra, 1994). Its objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of the product, independent from any particular system. STEP supports all branches, electrotechnic, mechanical and construction. The key features of STEP include:

- The use of formal methods for describing structure and correctness conditions for engineering information (STEP Tools Inc, 2001a).

- The definition of mappings from data models to implementation forms (EXPRESS-X).
- The definition of powerful resources for product modelling and in particular geometric modelling (Pierra, 1994).

In its development process STEP focused on explicit product modelling and in particular on explicit geometry. STEP was developed from a number of standards; IGES, PDES and DIN, and VDA/FS a US standard which demonstrated that basic geometric data could be exchanged between various drafting and modelling systems (Zeid, 1991). STEP is a set of ISO standards which provide for the exchange of engineering product data. These standards can be grouped into infrastructure components and industry specific models:

- A library of general purpose information models for aspects such as geometry, topology, product identification, dates, times. These are covered in the 40-series parts.
- Industry-specific application protocols that are built from the library of general models, the 200-series parts. Examples of these parts are: part 201 covering explicit drafting and part 202 covering associated drafting (STEP Tools Inc, 2001b).

The most important benefit of STEP is its extensibility. This is enabled by the language on which STEP is built. EXPRESS is a formal information requirements language (ISO/WD 10303-11:1998), which focuses on the definition of entities (objects of interest). Each entity is defined in terms of named attributes or properties, which are related to the entity and the representation, and relate to the specific domain or schema. Attributes related to entities may define geometry and behaviour: the events it responds to and how it responds.

The primary utilisation of STEP for manufacturing and production in Europe is the transfer of solids for digital mock-up, 78 percent of companies implement STEP for the transfer of solids and 12 percent for the transfer of wire frame models (PDES Inc, 2001). STEP has been successfully applied and developed for the design and manufacturing of sheet metal dies. The information models contain data on sheet metal part design data, sheet metal die and die set design. Although much of this data describes only properties and physical dimensions, and does not include any behavioural or performance characteristics.

Pierra (1994) attempts to extend the concepts of STEP by presenting a parametric product model for STEP. Although the model still consists primarily of geometric data is does enable assembly or whole part modelling. This is achieved by clearly separating the (public) interface of a parametric model and its private content. This interface consists of a set of (input) parameters and a set of (output) items which depend directly or indirectly on the values or parameters of other parametric models. The inclusion of such information permits the modelling of whole parts and assemblies whilst still providing encapsulation.

In contrast to bond graphs which lack certain geometric information, STEP covers in detail all the necessary geometric information for objects, and with the inclusion of parametric models assembly modelling is also covered. However, STEP lacks the capability to represent and manipulate the full range of performance attributes that are required for the sizing and specification of engineering components. Therefore, the use of STEP as a communication medium in a software environment for performance modelling and the embodiment of systems with standard components is inappropriate. However, as the usage of STEP increases, the outputting of assembly or component information in such a format might be considered.

3.2 Representations for engineering components

The importance of considering 'real' engineering components within a system modelling environment is discussed in chapter 2. For the purpose of this work, real components are those elements which may already exist, have been previously used and have predetermined or predefined properties. In many cases, these are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified. In order to provide for these 'real' components, the models which govern their design, selection and specification must be considered within the system modelling approach. Furthermore, section 2.3 discusses the benefits that arise from using the wide variety of electronic representations provided by third parties. This is because such third party representations are generally company and component focused, providing search algorithms and supplementary data which is important for the specification of a particular component type from a given manufacturer. As a consequence of this, there is a requirement for existing and emerging electronic representations which govern the design or selection of mechanical components to be incorporated within the modelling approach.

For the purposes of this work, a distinction is made between *models* and *representations*. There are a wide variety of modelling techniques in engineering design. These include process models, data models, information models and mathematical models (Court, 1995). For this work, *models* are the underlying principles or algorithms that represent a particular mechanical component and are termed component based models. These models are generally created from accepted scientific principles and provide for the sizing, selection and specification of the considered engineering component. *Electronic representations* are the software module(s) that encapsulate the model, and provide the support functions necessary for handling and manipulating the theoretical model in a software environment. These support functions provide for the visual interface, graphical support, search algorithms and user support functions, as well as access to the controls and object libraries contained within the operating system. Therefore, in order to incorporate third party models within the modelling environment it is necessary to either recode all the representations or integrate the various types of electronic representation in the modelling approach. The task of recoding these many representations is unviable due to the level of resources required. Furthermore, a distinct advantage of methods which incorporate third party representations is that no modelling procedures are prescribed *per se*, and therefore, do not impose any modelling analysis techniques on the designer. Indeed, the fact that many modelling environments impose modelling methods on the designer is highlighted by Subrahmanian *et al* (1993) as one of the most critical limitations of many approaches.

For the specification and design of engineering components there are many classes of electronic representation available to the designer. A study of the various types of electronic representation reveals nine key classes of representation:

1 Web based catalogues typically coded in HTML and using various scripting languages to access server-side databases.

2 CD-ROM software environments coded in BASIC, C or similar incorporating a database and advanced selection algorithms as well as supplementary technical data and images.

3 Standard representations such as STEP/PDES models in EXPRESS. Although typically these represent geometry and lack the performance data necessary to fully specify a third party component.

4 Parametric models for components and assemblies. Libraries are available for CAD systems such as SolidWorks (SolidWorks Corporation, 1998) and MechDesktop (Autodesk Inc, 1997).

5 Commercial analysis and design tools for standard designed components. These often follow standard design procedures and are becoming more widely available.

6 Enterprise product data management systems. These contain listings of available inventory and stocks for engineering components which may be consulted by the designer.

7 Custom codings for components that are regularly designed or procured. Often implemented in applications such as Microsoft Excel or Microsoft ACCESS and usually company specific.

8 Models of standard designed components such as those found in engineering literature for shaft design or gear design. These are often numerical codes written in C, Fortran or similar.

9 Sophisticated and specialised software for the design of 'one off' components using FEA, CFD, dynamic analysis tools or similar.

A set representation for the classes of engineering component and their various representations is depicted in figure 3.5. There is an overlap between electronic representations for standard selected components and standard designed components because standard designed components often utilise standard or preferred ranges for certain attributes. In addition to this, there is also a considerable overlap between bespoke designed components and standard designed components. This is due to the increased analytical capabilities of packages such as Microsoft Excel and also the increased performance of hardware and the ability of desktop PCs to perform complex analysis procedures, which can be invoked for standard designed components. In the context of this work, the figure demonstrates that if standard selected and standard designed components are to be incorporated in a modelling environment than a protocol for interfacing electronic catalogues, proprietary software such as Microsoft ACCESS and Excel, and CAD systems must be developed.

3.3 Current system modelling approaches in engineering design

This section appraises a number of system modelling approaches and corresponding computer based support tools in the field of engineering design. The approaches selected for review are those support tools that span the conceptual and embodiment stages of the design process, identified in section 2.7. This appraisal focuses on the methodologies for *representing a system*, *representing interactions* between elements in the system and the *analysis* or *resolution* of the system. For the purpose of developing a system modelling approach that represents the performance of mechanical systems for the task of system embodiment, four aspects of the various modelling approaches are reviewed.

1 Identification of the limits of applicability for the particular domain and for representing different component types.

2 The method or strategy for representing the system. This includes the types, arrangement and connectivity of elements within the system.

3 The protocol for representing interactions between elements. This includes the extents of interactions handled and the mechanism for the exchange or communication of data describing these interactions.

4 The strategy for analysing or resolving the system model. This aspect of the modelling approach deals with the determination of an acceptable solution or the evaluation of the performance of the considered system.

3.3.1 Evaluation of an assembly modeller for power transmission systems

Culley and Theobald (1997) describe a subassembly modelling tool for the embodiment of rotating power transmission systems. The work focuses on the configuration and optimisation of subassemblies from standard components and in particular the incorporation of standard catalogue components in the process. Their research established that the majority of engineering assemblies consist of three generic types of subassembly which are all interconnected (Culley *et al*, 1990). These generic classes are rotational, linear and combined motion elements, the former of which is the domain used for the development of a strategy for modelling subassemblies.

Overview

The methodology provides for both standard selected components and standard designed components. This is achieved through the creation of abstracted parametric models for components from catalogues and accepted design procedures respectively.

The modelling environment supports a schematic representation of the subassembly with individual components represented by icons and their arrangement determined by their location along the length of the core element, in this case the axial length of a shaft illustrated in figure 3.6.

Once the schematic representation has been configured, the designer may then enter any performance requirements by specifying desired parameter values for individual components. Following this, the model is resolved. This resolution process interrogates parametric models in order to determine feasible specifications for each component in the subassembly and thus a feasible overall solution. The approach also provides for limited optimisation of the mass properties for the assembly. The designer selects a range of driving parameters from different components, which are varied by the optimisation engine and the system resolved for each iteration. In this manner, an assembly of feasible component specifications which deliver the required performance with a reduced assembly mass can be determined.

Limitations of applicability

In the method described by Culley and Theobald (1997) engineering components are each represented by an abstracted parametric model. Consequently, the approach can only identify or suggest preliminary component specifications. This is because the parametric models are continuous representations of component ranges, and as a consequence, do not account for the characteristic that many standard components only exist over a finite number of discrete sizes. However, the limits of performance for a component range are captured within the model by bounding performance or physical parameters. If any parameter value exceeds these bounds the designer is notified and prompted to take the appropriate action. Because the approach implements these abstracted parametric models the methodology does not deal with real components. Furthermore, the designer may only utilise the limited range of component models included in the environment, additional models require abstraction and expert coding.

System representation

Individual assemblies are configured around a single core element. Sequences of components are linked together by means of nodal chains each connected to the core object. For rotating power transmission systems, assemblies are constructed around the shaft, illustrated in figure 3.6. These nodal chains are comprised of four object containers:

1. The nodal element is the locator for the component chain to the core object and determines the relative position of the nodal chain in the design model. And in this case its axial position along the shaft.
2. Auxiliary elements couple the application element to the core object and in the case of a gear pair would take the form of a keyway or spline.
3. Application elements describes the physical features of each design node and provide for engineering components which transmit power and forces. These include bearings, gears and v-belts.
4. External members provide the inputs and outputs to the design model and also a means to represent the physical characteristics of external elements, such as the support provided by a housing to a bearing or the power from a motor.

This format for system representation limits the scope of the system model to a single core element and hence only a single subassembly. Furthermore, the assembly may only contain a limited number of components, which when configured must be placed within containers for the particular class of element and as a consequence the approach assumes ideal inputs, when in reality subassemblies will be linked by sequences of mechanical components to form assemblies and systems (Hicks & Culley, 2000).

Representing interactions

An arbitrary assembly for a simple transmission is shown in figure 3.7. This figure illustrates the high level of data exchange necessary to describe the physical interaction between connected components in the assembly. To address this issue, the modelling approach exchanges a predetermined range of parameters between adjacent objects in each nodal chain. The transfer of these parameters is enabled by a nodal data field. This data field contains all the parameters necessary for all the various classes of object in the nodal chain, and is both written to and read from by each element in turn, according to the order of resolution. Each class of component in the nodal chain reads and consigns parameter values to a predetermined range of data addresses within the nodal data field. In this manner, the same data is always passed from the auxiliary element to the application element and from the application element to the external element, all of which pass through the nodal data field. Further to this, nodal data fields may also exchange parameters. This exchange is necessary for the resolution of the core element and the updating of parameters for all the nodal data fields.

System analysis/resolution

System resolution involves the interrogation of component models in a predetermined sequence. The order of this sequence ensures that all the data necessary for the execution of a particular class of object is available prior to the execution of the governing parametric model for the component specified in the object container. The propagation cycle of data through the complete design model incorporates three phases. The first phase of the process commences at the base of each nodal chain, i.e. the external element, and then propagates up the chain. Elements in each class are interrogated and the nodal data field is updated. Once all the elements in the nodal chains have been interrogated, data is propagated between the nodal chains and the governing model for the core object is interrogated. The final phase of the cycle is to resolve the nodal chains with the data from the core element. The nodal chains are resolved in a top down manner ensuring that the data from the core element is propagated through the design model. In this manner, design parameters specified in nodal chains are also propagated through the design model, albeit by virtue of the core element. This is essential so that speed, power and loading distributions are propagated though the system.

3.3.2 Evaluation of a fluid power simulation tool

There are a number of simulation tools for the dynamic analysis of fluid power systems such as BATH*fp* (Sidders *et al*, 1996) and AMESim (AMEsim, 2000). These tools aim to integrate the simulation process with the design phase of fluid power systems. In order to achieve this integration of simulation and design, the approach adopted is to simplify the simulation process. This simplification is achieved by the introduction of a mixture of modelling techniques to obtain representative system performance and by simulating purely dynamic analysis rather than steady state or frequency response analysis.

Overview

A schematic of the desired system is configured from the available component model icons and the appropriate ports are connected between elements to form the circuit. Once constructed, governing models must be assigned to each component, including connecting pipes and signals. These models are selected from the model library, which comprises component representations of varying levels of abstraction. This abstraction is dependent on the application, and in the case of a pump, models exist for both an ideal pump (100 percent efficiency) and a real pump, which accounts for losses.

The performance and physical characteristics for each component are then set, although each model has default parameter values, so that full data sets are always available. These parameter values are embedded into the component models and afford initialisation data for the simulation process. Simulation can be invoked once the required run time and data output intervals have been established. The latter of these enables the characteristics of the system to be graphed over time, an important feature for the analysis fluid power systems.

Limits of applicability

One of the most powerful features of these simulation tools is that they provide for the development of new models. This feature is enabled by a software tool that provides a template for model configuration. The first stage in the process is to create the schematic icon that will represent the component. The second stage involves defining the ports, their associated physical parameters and units, i.e. the physical quantities to be exchanged by the model. The second stage is to declare all of the model parameters and define their initialisation values, after which the governing computational algorithms must be coded, in either C or Fortran.

System representation

For the simulation tools appraised in this work the system or circuit is represented by simulation code. This code is generated from the schematic representation of the system model, and is comprised of elemental models encapsulated in functions and an associated call list, which executes the functions during the simulation cycle.

Representing interactions

For the simulation of fluid power systems, only flow rates and pressure are required to describe the interaction between the hydraulic components and represent system performance. This is shown in figure 3.8. These quantities are passed between arguments in the system code. Geometry and other performance characteristics are not required. However, it is essential that physical units are matched. To address this, simulation methods implement compatibility analysis. This analysis comprises two phases. In the first of these phases the connected ports of adjacent models are evaluated for physical parameter compatibility. This evaluation compares the permitted physical quantities that each model may exchange with connected component(s). The permissible transfer quantities for each component are determined during the configuration of component models and include pressure, flow rate, force and torque. The second phase of compatibility examines the units of measurement associated with all connected ports. These physical units are also determined during the configuration of component models. Therefore, for coupled components to be compatible the physical quantities and units of measurement must match across their interface.

System analysis/resolution

The simulation code comprises a call list for the component models and declares all the system variables. The sequence of the call list is primarily determined by the order in which the system model is configured, such a scheme is adequate for what is essentially closed loop system analysis. This is because for closed systems a sequential operation will eventually consider all the elements, regardless of the initial start position. However, there is a prerequisite for some components to be resolved after others, and in the case of fluid power circuitry these are pressure dependent models, such as pipes. Hence, pipe models cannot be explicitly resolved until the flow across them has been established. As a consequence, the precedence incorporated into the modelling approach is to firstly resolve algebraic models, followed by models that contain integrator elements.

3.3.3 Evaluation of an electrical circuit simulator

For the purposes of analog circuit simulation the majority of commercial tools available implement a modelling methodology based on or compatible with the SPICE method (Keown, 1994). The initial development of the SPICE method is attributed to the University of California, Berkley and implements a syntax that describes the types of elements in the circuit and the analysis to be performed (Kielkowski, 1994).

Overview

For electrical circuit design the majority of simulators provide an integrated design environment (IDE) comprising a schematic circuit designer, a SPICE simulation engine and a printed circuit board (pcb) layout designer (MicroSim Corporation, 1997).

Construction of the system model involves editing a schematic of the circuit. This includes the selection of electrical parts, connecting wires, buses, ports and external connections. All parts, ports, buses and wires have associated attributes which may be edited by the user. These attributes are used for the construction of a circuit file (netlist) and later in the design process to produce a bill of materials. Once configured, the circuit model may then be simulated for AC or DC analysis, transient analysis and sensitivity analysis. After simulation the current and voltage characteristics for all elements may be displayed as traces. Following a successful simulation, the user may then use the pcb layout tool to place parts and connectors, perform packaging design and generate a bill of materials.

Limits of applicability

Because of the wealth of tools for electrical circuit simulation and the similarity of the underlying simulation engines there is a vast range of compatible part models available. Many of these are contained in the parts libraries from the software providers or readily available from suppliers, manufacturers and academic institutions. In addition to this, the simulation environments often comprise a model editor which can be used to characterise specific models from data curves or specify models from certain vendors (Silvaco International, 1999). Furthermore, the environments also provide for blocks. These hold a collection of circuitry, the behaviour of which is defined by the user. The system treats these as a 'black box' and does not perform any checks on these elements.

System representation

The system model can be simulated for AC or DC analysis, transient analysis and sensitivity analysis. Prior to the simulation episode an Electrical Rule Check (ERC) is performed on the model. This checks for open input pins or conflicting outputs. If the ERC is successful then a netlist is generated. The netlist or circuit file describes the connectivity of the circuit including all the components, their interconnections and their values. In addition to the netlist, the simulation engine requires the template for each component. The template attribute specifies the contribution of primitive parts and uses the PSPICE simulation netlisting syntax. In the process of creating the netlist; buses, connectors and parts are resolved, and only elements with a template attribute are included in the simulation episode.

The netlist file conforms to the PSPICE syntax, and is in an ASCII file format. Essentially a text file which lists the connections of a schematic, by naming the connected signals, parts and pins, and associated values.

Representing interactions

For the purpose of simulating electrical systems, the behaviour is represented by two design parameters; voltage and current with respect to time. In such an approach, information is not explicitly exchanged between models, rather the voltage and current across each component and at each junction are evaluated. For each junction the relationship between current and voltage is represented using Kirchoff's 2nd law, depicted in figure 3.9. This generates a set of nodal equations for each junction. These equations comprise behavioural models for each component located along the paths that form the junction. An example of a behavioural element for a nonlinear diode is $I_d = I_s\left[\exp\left(\frac{qV_d}{NKT}\right) - 1\right]$ (Keown, 1994). Once a full set of system equations has been constructed then a solution state can be determined.

System analysis/resolution

In order to perform the simulation, a netlist must be compiled and a command file containing the simulation commands and specifications of each model library must be generated. Once the circuit netlist has been compiled then the circuit parameters may be varied, these include for example temperature and humidity. The first phase of system resolution is to decompose the model into a system of nodal equations. This is achieved by considering the current leaving each of the circuit nodes, performed using Kirchoff's laws. The resulting nodal equations are transformed into a set of system equations and then represented in a set of matrices. To solve the system matrix, SPICE implements two solution algorithms; one for linear elements and one for non-linear elements. For linear analysis the computational equivalent of Gaussian elimination is used and for non-linear elements the Newton-Raphson algorithm is used and elements are broken-down into small linear approximations.

The SPICE simulation engine treats components as if they were ideal non-interacting elements, where this interaction refers to the production of electrical field effects such as latch-up. Circuit blocks are represented as behavioural elements and simulated in a functional form.

In real life, the performance capabilities of electrical components can vary by 1 percent to 5 percent of that specified. Consequently, designers try to develop circuits that are insensitive to these variations and many vendors offer 'Monte Carlo' and 'worst case' analysis which automatically varies circuit components and performs multiple simulations to simulate these variations and evaluate the resulting performance.

3.3.4 Comparison of design tools for engineering domains

The previous sections evaluate system modelling tools from different engineering domains where such tools are now widely accepted as key support methods or design aids. The critique of the modelling strategies is undertaken in order to generate an understanding of the key aspects to a systems modelling approach and discuss the various methods with respect to the development of a strategy for modelling mechanical systems. The critique focuses on three aspects of the modelling approaches; *representing a system*, *representing interactions* and *analysing the system model*. Further to this, the general features and limitations of each modelling approach are discussed and summarised in figure 3.10.

System representation

In all of the modelling approaches reviewed in this work, a system representation is constructed which describes the relative arrangement and connectivity of elements within the system. For simulation tools in the fluid power domain, this process is simplified by the fact that the many systems form what is essentially a closed sequential loop for the purpose of analysis. Although in the circuit schematic parallel elements or paths are represented by the inclusion of valves and pipes as components. This is also the case for electrical circuits, although parallel elements and hierarchies are more prevalent than in fluid power systems and their analysis is simplified by the fact that the nodes or junctions of the circuit are used as local points of reference during simulation. In contrast to these relatively sequential system representations, mechanical systems generally form a complex network of elements, illustrated in figure 3.11, where components may well be connected to many other elements which all interact. This complexity requires a more involved representation which captures all the relationships. In the case of the assembly modelling tool reviewed in section 3.3.1 the form and topology of the representation is predetermined, such that only a single assembly of a limited size can be modelled.

Representing interactions

System modellers for the design of fluid power systems and electrical circuitry require only two parameters in order to describe the interaction between elements and evaluate the system performance. These are the propagation of information describing flow and pressure and the summation of voltage and current characteristics respectively. However, for a mechanical system modeller there is a requirement for many more parameters to be exchanged. Figure 3.7 illustrates the information requirements between components in a simple transmission system. Here it is evident that an extensive and varying range of component information is required by adjacent components in order for their resolution. This specification is also complicated by the fact that much of the information may need to come as the result of selection from a third party catalogue or sizing from standard design equations which are external to the system representation. For the assembly modelling tool reviewed herein the problem of complexity of data exchange is overcome by the implementation of a predefined structure and a corresponding fixed level of communication between components within this structure.

System analysis

For system modelling tools in the fluid power domain, data propagation and model interrogation is sequential, although there is a precedence imposed for certain model types to be resolved last. These are the pressure dependent models. For mechanical systems, the order or precedence of data propagation and model interrogation is not as straightforward. Certain principal components, such as a shaft, may possess many connecting components which provide multiple inputs and outputs. Because of this additional complexity, there is a precedence for elements that provide inputs to the considered first. The principal component may then be resolved, following which components that convey outputs can be resolved. In the electrical domain the strategy adopted is a simultaneous resolution of the system model, in order to determine a solution state. This is feasible because behavioural models for components can be easily defined parametrically, where as for mechanical components the governing models may be implicitly defined in third party representations. The nonholonomic nature of these representations requires them to be accessed individually. Furthermore, the lack of causality within a mechanical system model hampers the ability to solve simultaneously. In the assembly modelling tool for power transmission systems, this is overcome by imposing a limited and predefined assembly structure. Consequently, the arrangement is always known and can be resolved in a predetermined manner.

3.4 Concluding remarks

This chapter has provided a review of current approaches and technologies for modelling mechanical systems and representing mechanical components. In particular, three areas are dealt with that impact on this work; methods for representing engineering systems, representations for engineering components and current system modelling approaches in various engineering domains.

The review of representations for engineering systems focuses on bond graphs and various product data models and in particular STEP. The review demonstrates that both of the approaches lack the ability to represent the extensive level of performance and geometric data necessary for the modelling of standard components for their design and selection within a systems modelling approach. Furthermore, the extension of these approaches to include the necessary range of data is not feasible, due to the levels of complexity that would have to be introduced.

The review of current technologies for representing individual mechanical components, highlights the wide variety and diversity of models and corresponding electronic representations. This variety and diversity is necessary to provide for the varying levels of complexity and analysis involved in the design and selection of different mechanical components. These may include databases, advanced search techniques, analysis procedures and mathematical modelling. Many of these representations have been developed overtime and are very powerful tools for the selection of individual components. As a consequence, it is desirable for such representations to be considered within a systems modelling approach so that 'real' components are dealt with. This issue is a particularly important consideration in the development of a mechanical systems modelling tool.

The final section reviews a number of system modelling approaches in different engineering domains. This review deals with three aspects of each modelling approach and in particular; *system representation*, *representing interactions* and a procedure for *system analysis* or *resolution*. The modelling approaches for the design of fluid power systems and electrical circuits have been successfully implemented and are now widely used as design aids. However, for mechanical systems modelling only domain specific support tools are available, such as the modelling tool for a single rotating power transmission assembly.

One of the main reasons for this is that mechanical systems are geometrically constrained in terms of their arrangement and physical connections over three dimensions. This means that parameters describing not only energy transfer but also the interfaces themselves must be conveyed. In contrast, in the fluid power and electrical domains only two parameters and one dimensional interaction are necessary to simulate system performance.

In addition to the issue of representing interactions, the structure of a mechanical system can be far more complex than either a fluid power circuit or electrical circuit, which can be represented as essentially closed loops for the purpose of analysis. Mechanical systems generally form a complex network of elements connected by principal components that possess multiple connections, such as a shaft. These two aspects add considerable levels of complexity, which must be addressed by a strategy for transferring and handling the interactions necessary to analyse the system model. In conclusion, it is necessary to develop an entirely new modelling approach for mechanical systems. This includes a strategy for system representation, a protocol for representing interactions and for system resolution. In addition to these aspects, a general review of the features implemented in the various modelling approaches highlights two support features necessary for successful system resolution. These are data arbitration and compatibility analysis discussed in chapters 5 and 6 respectively.

IDEAL ELEMENTS IN THE ENERGY MODEL						
Transfer elements			Energy sources	Energy stores		Energy dissipators
Conserve energy and transform effort and flow variables without any loss			System inputs either effort or flow	Either accumulate the effort or flow variables		Flow and effort co-variables possess the same relation
Junction		Transformer		Effort	Flow	
One of the co-variables must equal zero		Co-variables are transformed				
Effort	Flow					
$e_1 = e_2 = \ldots = e_n$ and $f_1 + f_2 + .. + f_n = 0$	$f_1 = f_2 = \ldots = f_n$ and $e_1 + e_2 + .. + e_n = 0$	$e_2 = ke_1$ $e_1 f_1 = -e_2 f_2$ where k is the transformer ratio, e_1 the effort in and e_2 the effort out	$e = e_0$ and $f = f_0$ where e_0 and f_0 are system constants	$e = {}^{q}/_{C}$ linear case $e = \phi(q)$ non-linear case where C is the capacitance ,φ(q) $q = \int f \, dt$	$f = {}^{p}/_{I}$ linear case $f = \phi(p)$ non-linear case where I is the inertance φ(p) $p = \int e \, dt$	$e = \phi(f0)$ $f = \phi^{-1}(e)$ nonlinear case $e = Rf$ $f = e/R$ linear case where R represents an electrical resistance $E = \int f^2 R \, dt$ $E = \int {}^{e^2}/_{R} \, dt$

Figure 3.1 – Ideal element models for bond graph construction (Gawthrop and Smith, 1996)

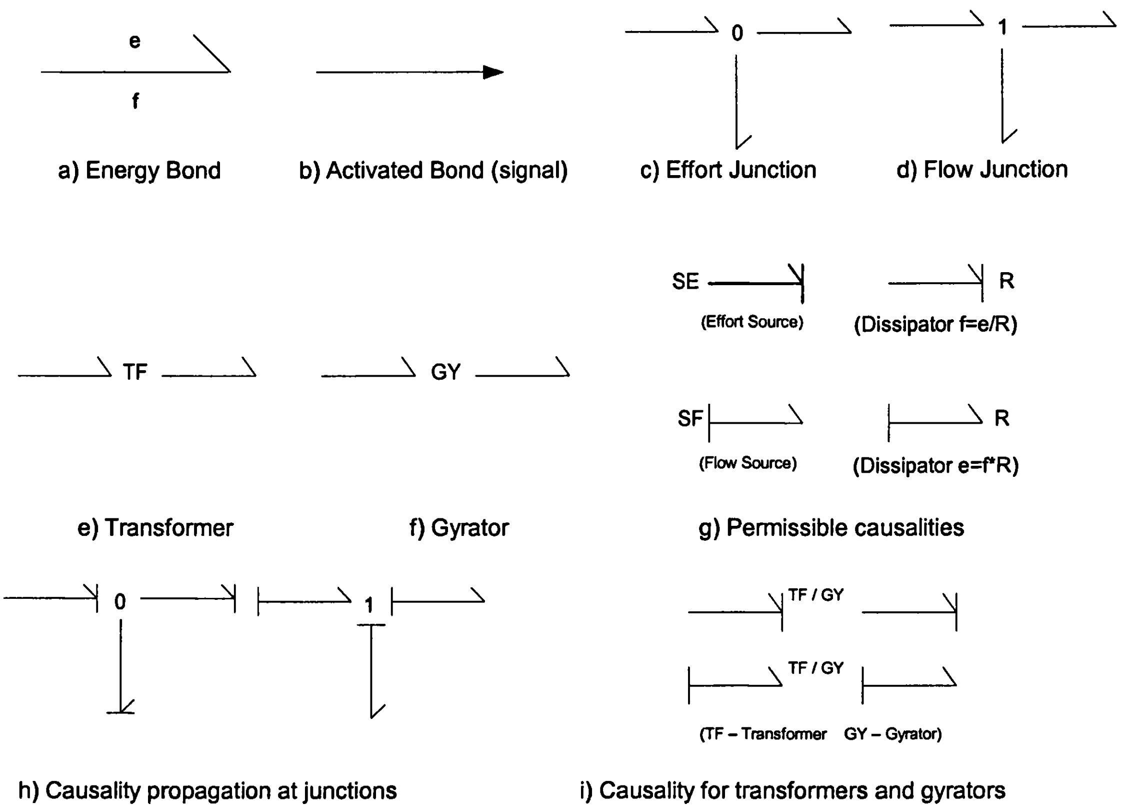


Figure 3.2 – Representation of bond graph signals (Gawthrop & Smith,1996)

Classification of variables for bond graph construction							
Domain	Effort (*e*)	Flow (*f*)	Momentum (*p*)	Displacement (*q*)	Flow store (c)	Effort store (I)	Dissipator (R)
Mechanical (Translational)	Force (F)	Velocity (V)	Momentum (p)	Displacement (x)	Potential energy (Spring)	Kinetic energy (Mass)	Friction
Mechanical (Rotational)	Torque (T)	Angular Velocity (ω)	Angular Momentum (h)	Angular Displacement (α)	Potential energy (Spring)	Kinetic energy (Mass)	Friction

Figure 3.3 – Bond graph variable classification for mechanical systems (Gawthrop & Smith,1996)

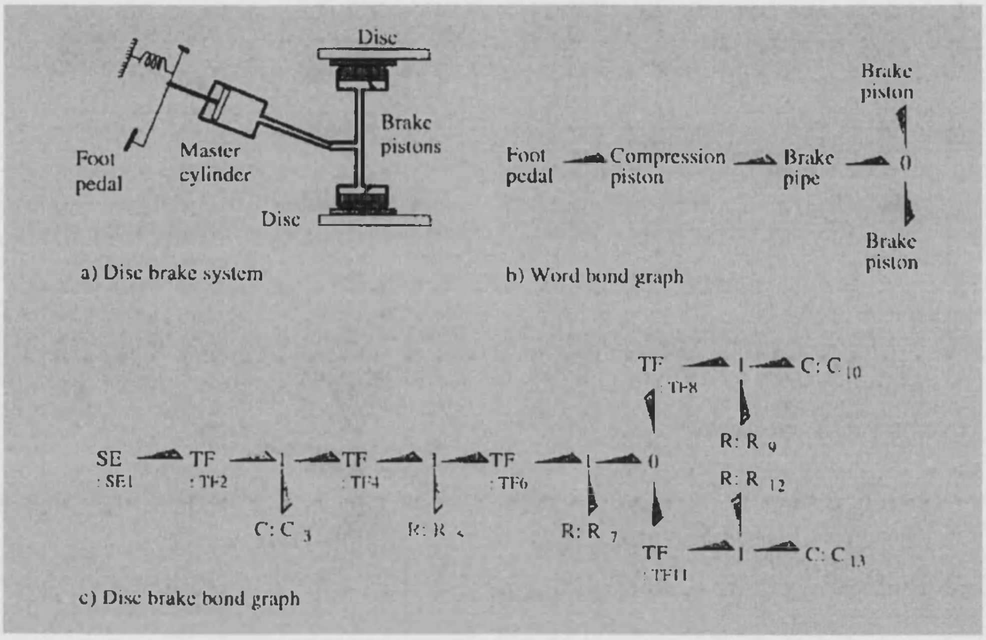


Figure 3.4 – Bond graph representation of a disc brake system (Gawthrop & Smith, 1996)

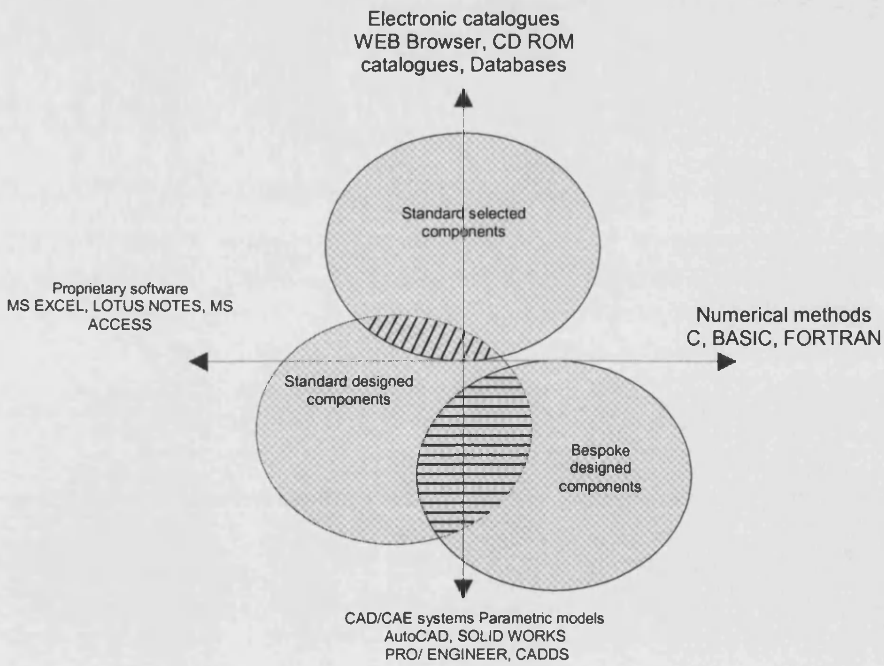


Figure 3.5 – Classes of engineering component and their associated form of electronic representation

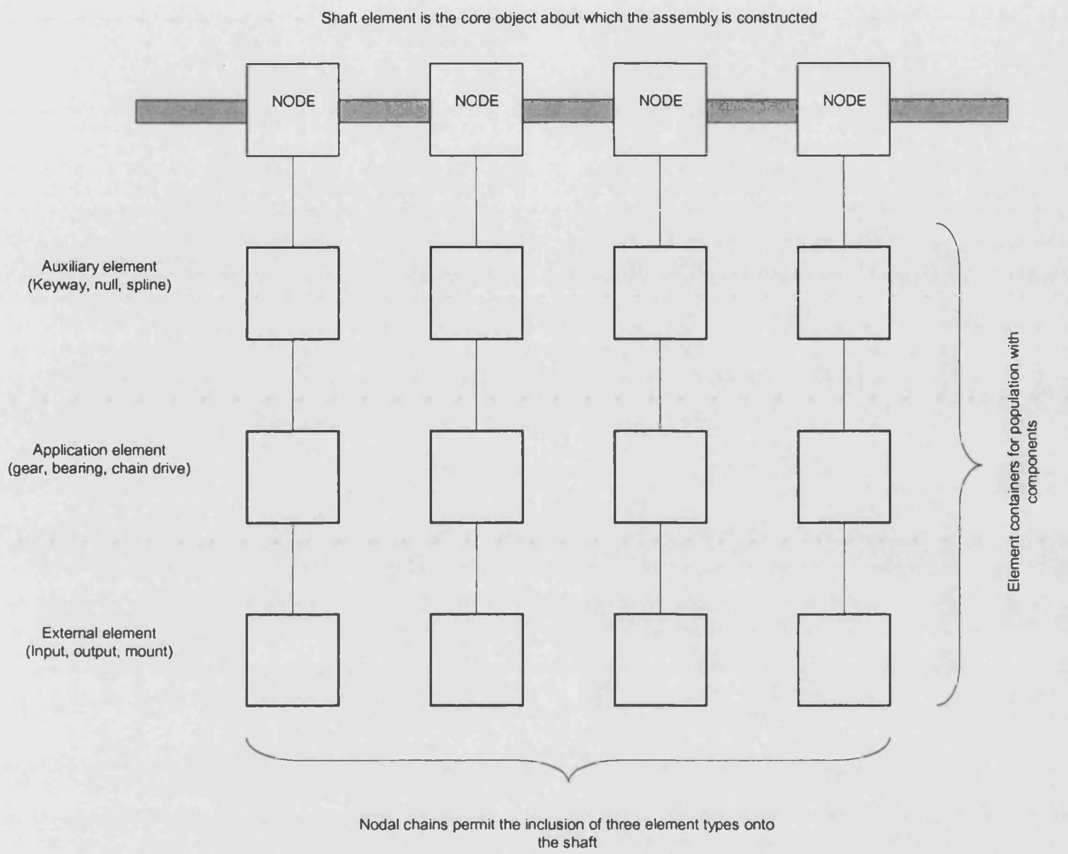


Figure 3.6 – Predefined structure for an assembly model

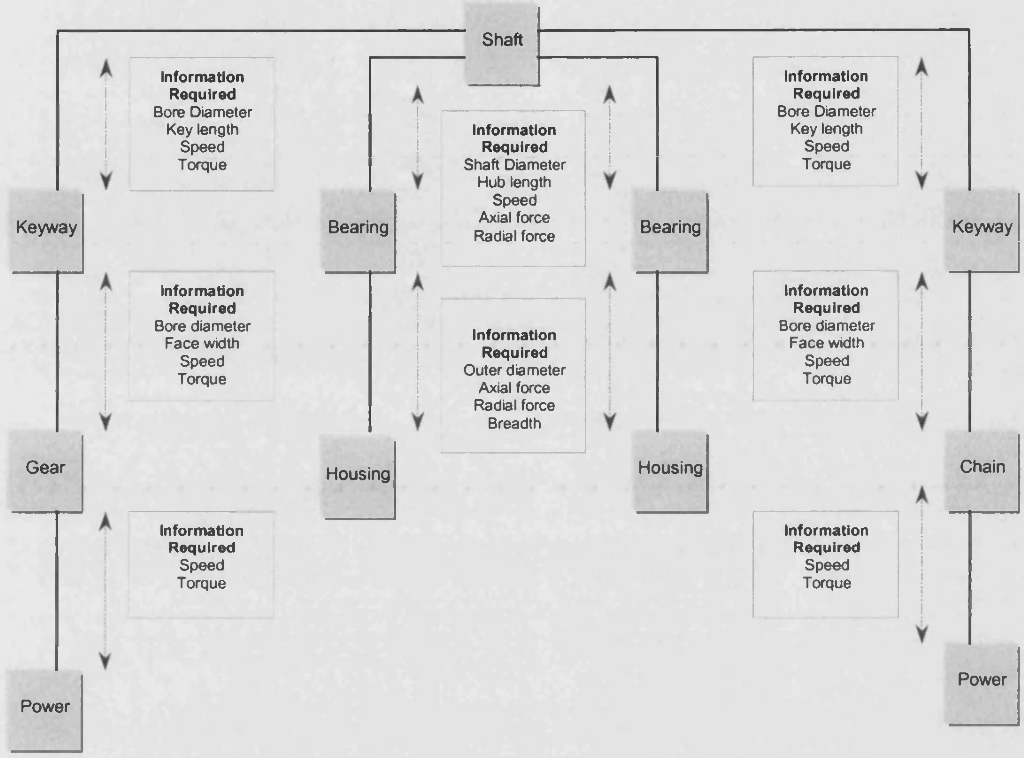


Figure 3.7 – The required level of information exchange between components in a transmission

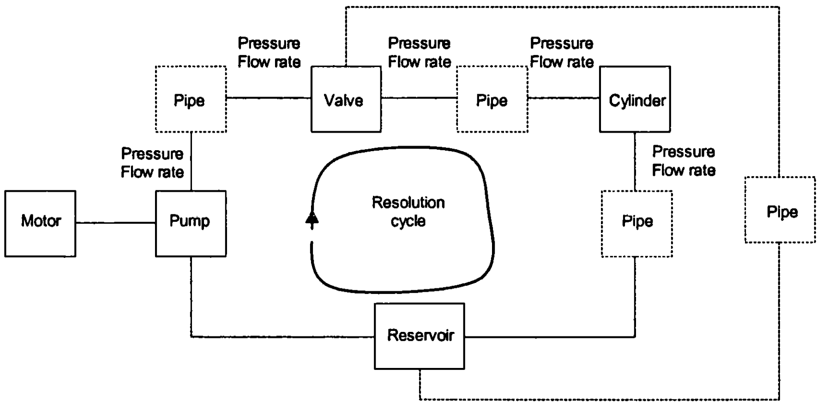


Figure 3.8 – The order of resolution and direction of data propagation in a fluid power circuit

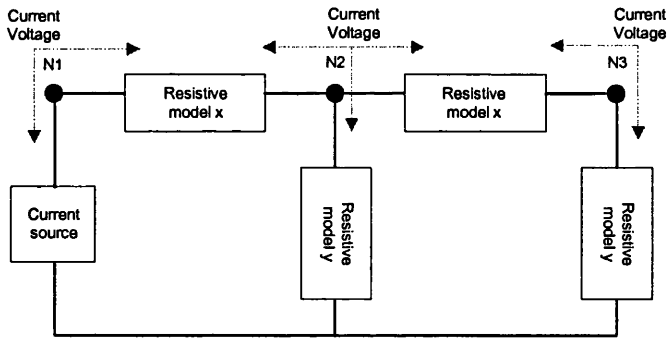


Figure 3.9 – The order of resolution and direction of data propagation in an electrical circuit

Features / Engineering Domain	Rotating Power Transmission Systems	Fluid Power Systems	Electrical Circuits
Model construction	Schematic design in iconic chains	2D schematic page	Flat or hierarchical schematic pages
Consideration of Geometry in system model	Leading Component dimensions determined	-	2D Layout for printed circuit board design
Production of 3D information/representation	-	-	-
Provision for catalogued components	Models derived form catalogues	-	-
Inclusion of standard components	Standard stress-analysis design models (shafts, keys etc)	Pumps, pipes motors	Design procedures for electrical elements
Production of component selection data	Preliminary component sizing	Performance data and sizings	Exact specification of components
Generation of Bill of Materials/parts list	-	-	Bill of materials can be generated from printed circuit board designer
Component compatibility analysis or design advisor	-	Physical and units check for mated elements	Electrical Rule Checking (ERC) for constructed circuit
Optimisation functions	Power transmissibility and mass	Optimise for desired performance characteristic of system (loading/time)	Optimise for desired electrical performance characteristic
Conflict resolution/arbitration (eg user data v's calculated data)	Resolution of component conflicts eg sizes	-	-
Tolerance/sensitivity analysis of design model	-	-	Worst case testing for system with component tolerance
Parameter summing/global design attributes	-	-	Sum costs etc from Bill of materials
Representation at an assembly/sub-assembly level	Maximum predetermined no of elements	Any sized system	Size of schematic page
Representation of systems (multiple assemblies)	Single assembly only	Multiple systems	Flat or hierarchy of schematics can be created
Data passing between elements of the model	Hardcoded blackboards	Arguments in the simulation code	Component templates and attribute values passed into system equations (matrix)
Resolution (propagation of data and constraints)	Interrogation of component models and data fields along iconic chains	Simulation code created which calls pressure dependent models last	Solution of system equations (cpu equiv of Gaussian elimination)
System Representation	Data fields and ID numbers	Simulation code (C)	Circuit file or netlist in predetermined syntax
Model library	Object palette – collection of components and model	User selects a model from the library for the component	Global library – user selects models for each element type
Model editing facilities	-	BathMAT – used to create component model and icon	Existing models and parameters can be edited to suit a manufacturer
Model parameter editing	Component viewer – user specifies desired values	Parameter definition mode – user specifies values	Template attributes are specified eg resistance
Connectivity of models with the system	Hard coded into software	Input/Output parameters declared in and called by the system	Models and templates are coded in PSPICE syntax
Model types implemented in the system	Parametric	Parametric	Parametric
Introduction of UDF Models	-	Using BathMAT	Generally models are (developed by a manufacturer or academic institutions)
Modelling Language	C	C/Fortran	C
Software Environment	Windows	various	various
Drag and Drop	Icons selected from palette and placed in iconic chains	Symbols and objects placed on worksheet	Symbols placed and drawn on schematic page
Software Connectivity	-	-	Compatible with SPICE simulators
Coding Requirement	-	C or Fortran for adding models	-

Figure 3.10 – Comparison of modelling approaches in different engineering domains

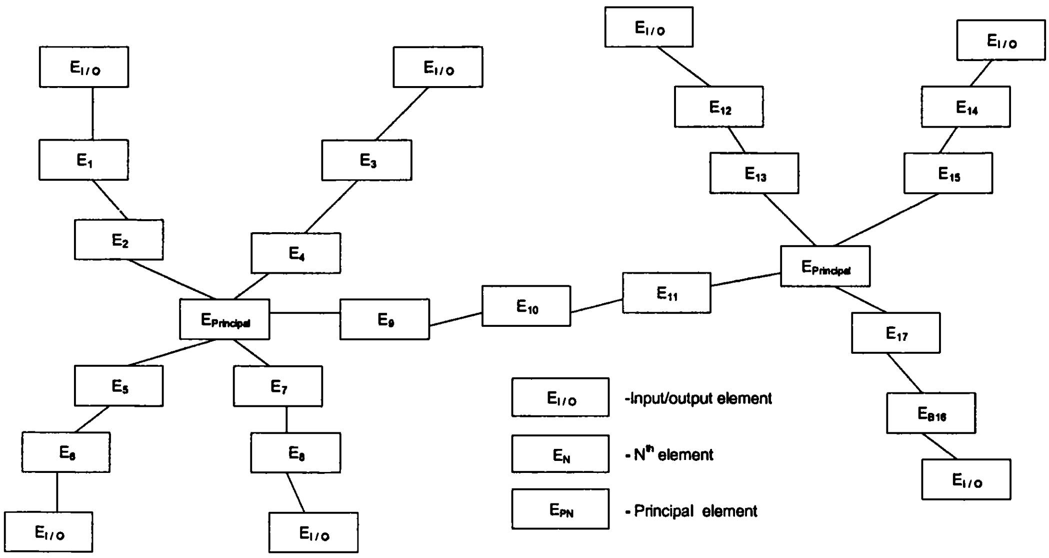


Figure 3.11 – Arrangement and connectivity in a mechanical system

Chapter 4

A system modelling approach for machine systems

In chapter 3, various techniques for representing mechanical components and systems are reviewed. This review also appraises a range of system modelling approaches for various engineering domains. This critical review highlights the inability of current technologies and modelling approaches to provide for a mechanical *systems modelling* approach that incorporates *electronic representations* of standard mechanical components for machine systems design. As a consequence of this lack of support in this important area, there is a need to develop a new systems modelling approach for mechanical systems and in particular machine systems. In the development of this approach there are a number of key issues that must be addressed. These include dealing with the complexity of mechanical systems, the high levels of data exchange and the complex order of data exchange necessary for system resolution.

1 *Dealing with the complexity of mechanical systems.* A mechanical system may consist of any number of components connected in a variety of complex configurations. This variety and diversity are addressed by a strategy for system representation, which describes the arrangement and relative connectivity of elements within the system.

2 *Providing for the high levels of data exchange.* One of the fundamental issues to be dealt with is the wealth (number) and diversity (types) of performance data and geometric data that must be communicated or exchanged between elements in the system. This is necessary in order to match components with respect to their performance capabilities, and design and select components that possess a synergy effect that achieves the desired overall performance characteristics for the design. This is dealt with in the development of a strategy for representing interactions.

3 *The complex order of data exchange.* The ability to consider systems and networks of standard mechanical elements as a whole demands a procedure for controlling the exchange of data within these complex structures. This aspect of the modelling approach is addressed by a strategy for system resolution. This strategy also manages the order in which component based models (electronic representations) are interrogated. These two features

ensure that all necessary data required for the interrogation of component models is available, component models are interrogated at the correct stage in the process and that data produced by component models is available to other system elements.

The three key features of a system modelling approach; *system representation*, *representation of interactions* and *system resolution* are developed in the following sections.

4.1 System representation

In order to effectively model a mechanical system, a representation needs to be constructed which describes the relative arrangement and connectivity of the elements. For mechanical systems, this arrangement can be a complex network where components may be connected to many other components. An example of an arbitrary assembly is illustrated in figure 4.1. Even in this simple example the structure of the system is relatively complicated, comprising sequences of components which emanate from a number of core components. In order to represent systems of this complexity an approach that is based on <u>connections</u> rather than <u>components</u> is developed.

This strategy has been developed through the evaluation of a range of mechanical assemblies and considers the order of connectivity within a system model. Through the consideration of the number of connections which relate to a particular element it is possible to differentiate key components. For the purpose of this work, key components are the system inputs, system outputs and core components. Core components are the components from which all other component sequences emanate and provide the basis for all mechanical systems. In the case of rotating power transmission systems these core components will typically be shafts.

For the purpose of system modelling, each mechanical component can be considered to represent one element of the model. The term element is therefore used in the context of system modelling and component is used to refer to the physical system. In the system model key elements are classified according to the number of connections which they possess. This classification includes three classes of element; unitary elements, binary elements and principal elements.

1 *Unitary elements* possess only a single connection. This characteristic is only exhibited by the marginal or boundary elements, and it is the boundary elements that provide the inputs and outputs to the considered system.

2 *Binary elements* possess two connections and in mechanical systems are the components which convey the inputs and outputs to the core components, or link core components to other core components.

3 *Principal elements* possess more than two connections. These principal elements are the elements about which all other sequences of elements emanate. This class of element represents the core components in the mechanical system.

The adoption of such an approach enables the representation of system inputs and outputs, principal elements and the relative structure or arrangement of system elements. This is because each connection also represents the elements between which it is connected. Therefore, each connection and its relative position between unitary elements and principal elements can be determined. The proposed approach is applied to an example system in figure 4.2. It is possible to identify inputs, outputs, core components and the arrangement, in terms of connectivity, between each component.

4.2 Representing interactions

In order to effectively model a mechanical system there is a requirement for many parameters that describe the physical interactions between elements to be exchanged. This is particularly the case when compared to modelling approaches in the fluid power and electrical domains, as discussed in chapter 3. Figure 3.7 illustrates the necessary level of communication between components in just a simple transmission system. Here it is evident that data which describes an extensive and varying range of component attributes is required by adjacent components in order for their effective design and specification. This design and specification is also complicated by the fact that much of the information regarding the attributes of components may need to come as the result of the interrogation of an electronic representation, such as the selection from an external catalogue or sizing from standard design equations.

This section evaluates the extents of data demanded by mechanical components for their design and selection *a priori*. From this, a classification of component attribute types is generated. This classification is used to derive the necessary range of data to be propagated within a system model, such that the design and specification of standard components during the early phases of design can be undertaken. In particular, the level or extent of data exchange must be sufficient so as to enable the effective execution of the various classes of electronic representation.

Figure 3.7 illustrates that sequences and nets of mechanical components demand information about their physically linked components in order for their individual specification and ultimately the complete embodiment of the considered system. To address the issue of data exchange, a communication protocol is developed which enables the exchange of essential data. This data comprises a range of parameters which describe all the data required by a component from its connected components in order for the determination of related component attributes and the performance data necessary for component design or selection.

In the development of a communication protocol it is important to understand the relationship between parameters and attributes, this is shown in figure 4.3. An attribute may either be explicitly defined by a parameter or derived from a combination of parameters and/or other component attributes. A parameter is defined in the Collins Concise Dictionary (1990) as *"an arbitrary constant that determines the specific form of a mathematical equation"*. For the purpose of the protocol being developed in this work the mathematical equation may be considered to describe a component attribute. In which case, the mathematical *equation* which determines a component *attribute* is formed from *parameters*. Therefore, these parameters need to be communicated within a mechanical systems modeller so that the full range of component attributes may be determined. In order to identify which parameters to communicate, the dependency of component attributes on parameters must be assessed. Attributes are defined in the Collins Concise Dictionary (1990) *"belonging to, produced by or resulting from"* and are the component characteristics on which the selection and specification of a real component is based.

4.2.1 Attribute classification for mechanical components

In order to determine the set of parameters which must be communicated it is firstly necessary to investigate component attributes and to determine their reliance on the system and other components in the system. The objective of this approach is to provide sufficient data exchange for the generation of component attributes that are driven by other connected components. To consider the full range of parameters, attributes have been investigated for a range of standard designed and standard selected components. This investigation reveals that attributes may be classified into three-tiers. These tiers or classes differentiate attributes according to their method of formulation and relates their dependency on system attributes, attributes that are specific to a particular component and the attributes of other system components, with particular attention to physically linked components, as described below.

Global attributes

These are required or specified attributes which are consistent for all components in the system. Such attributes may include working environment, required life, duty cycle and fluid types for hydraulic systems. Global attributes need to be defined at a system level, or if multiple assemblies are to be modelled then a set of global attributes may need to be declared for each assembly.

Local attributes

These are component attributes that are determined in part, or full, from parameters that are driven by physical connections with other components. An example of such an attribute is the dynamic load rating of a bearing, which is dependent on the radial forces, axial forces and rotational velocity of the shaft. For this case, the various attributes of the shaft constitute parameters for the determination of the required dynamic load rating of the bearing.

Intrinsic attributes

These attributes are specific to a component type and size. Often such attributes may be neither described mathematically by accepted algorithms, nor directly dependent on the connected components or the system. However, they may be indirectly related to other attributes and attribute types. For example, the intrinsic attribute mass is dependent on the component dimensions and material attributes. Therefore, the mass attribute is indirectly related to other attributes. However, this indirect relationship cannot be readily described mathematically and will typically have been determined empirically by the manufacturer. As a consequence, the designer will normally source such attributes from a catalogue or other manufacturer's information.

Example implementations of the attribute classification for a standard shaft coupling and a gear pump (RS Component Limited, 1997) are detailed in figure 4.4. Having identified the key component attributes which demand the communication of parameters from connected components, i.e. local attributes, it is possible to group them into four quantitative classes. This overall attribute classification is depicted in figure 4.5.

4.2.2 Local attribute subgroups for mechanical components

In order to achieve the objectives of a communication protocol it is necessary to further subclassify classes of attributes into characteristic local attribute types. The proposed framework for this subclassification of local attributes is as follows:

- *Form/geometric* attributes are characteristic attributes which describe the arrangement of surfaces (or spaces) for the connections or interfaces between components.
- *Physical* attributes characterise effects which can be described quantitatively by means of the physical laws governing the physical quantities involved (Pahl & Beitz, 1996).
- *Motion* attributes describe changes in the physical position of a component or the connection/interface and include rotational and translational quantities.

- *Other* attributes encompass those subclassified attributes that cannot be directly classed in any of the previous groups but may still describe physical effects and or quantities. These quantities may include parameters to form an optimisation function or goal, such as mass, cost, spatial occupancy or power transmissibility.

4.2.3 Global and intrinsic attribute subgroups for mechanical components

In addition to the sub-classification of local attributes, both the global and intrinsic attribute categories may be grouped into two distinct subclasses; deterministic and specified attributes. The significance of which, is that attributes in the former class are typically dependent on the attributes of a component whilst the latter are usually independent although they are dependent on a component type[1] or range (range dependent).

- *Deterministic attributes* are driven by changes in a component's characteristic attributes, i.e. leading dimensions. Such attributes are mass, cost and internal clearances which are all properties of a component's physical characteristics.
- *Specified attributes* are those selected by the designer from the included range for the respective component. Examples of such attributes include lubrication, shields and seals which may only take the value of the available options for the component, i.e. for lubrication these might be either oil or grease. These are not only textual values, some maybe numeric and determined empirically by the manufacturer such as the angular misalignment for a bearing.

The proposed attribute classification and subclassification scheme is illustrated in figure 4.5 and its application to the extensive attributes of a cylindrical roller bearing (SKF Limited, 1997) and a Uni-lat coupling are depicted in figures 4.6 and 4.7 respectively.

4.2.4 A protocol for communication in a system modelling environment

The three-tier classification scheme proposed for global, local and intrinsic attributes provides a framework that groups the attributes of a mechanical component according to their dependency on the attributes of the system and those of other physically connected components. The objectives of this classification are to determine the range of parameters necessary to enable the generation of local attributes, and to differentiate the required mechanisms of communication for

---

[1] For the purpose of this work, a component type defines a particular class of mechanical component from a specific manufacturer or supplier.

different component attributes and parameters in a systems modelling approach. Hence, each of the three classes demand varying degrees of communication within the system model and between connected elements, i.e. whether communication is required at a system level or at an elemental level. The different communication requirements for attribute classes are illustrated in figure 4.8 and discussed in the following sections.

Global attribute communication

Global attribute declarations are required to be available to all components in a system or assembly. If multiple assemblies are included then global attribute declarations for each assembly must be defined. In order to achieve this, a '*global whiteboard*' is proposed. The whiteboard in this context can be considered to be a knowledge depository which includes all system attributes, such as operating conditions or life. It is characterised by its read only status, that is to say information is available to system elements but may not be altered by them. An example of the format of a global whiteboard is illustrated in figure 4.9.

Local attribute communication

The local attributes of a component are determined by parameters that are exchanged with connected component(s). It is therefore necessary for these parameters to made available between connected components. This is achieved by the association of 'local blackboards' with each connection, see figure 4.8. The blackboards are formatted to contain subsets of parameters which correspond to the subclassification of local attribute types defined in the section 4.2.2, i.e. *geometric*, *physical*, *motion* and *other*. The blackboard is characterised by its read and write status. System elements interface with the blackboard to acquire information, following which elements may then consign information to the blackboard. The format of the 'local blackboard' is depicted in figure 4.9. The various classes of parameter and associated identifiers are described below.

- *Geometric:* These are the parameters that enable the positioning and representation of components in three-dimensional space. In the development of a modelling approach that supports the early stages of design it is not feasible, or necessary to include the level of information required to represent complex geometry. The set of geometric parameters proposed ensures sufficient information to enable the positioning of an object in three-dimensional space, the representation of its spatial envelope and connectivity surfaces or interfaces. This spatial envelope approximates objects to simple solids, i.e. cylinders, cubes and toruses, which is sufficient in order to define the arrangement of mated surfaces.

- *Physical*: This class of parameter includes the two physical effects that are necessary to describe the interactions between connected mechanical components. These are forces and torques respectively. It is postulated that all other relevant physical quantities may be derived from combining parameters in the *physical* and *motion* data packets, and magnitudes determined by using parameters from the *geometric* set.
- *Motion*: These parameters represent the complete set of positional changes, and include both translational and rotational velocities with reference to a Cartesian coordinate system. This is necessary so as to enable the inclusion of both linear, rotational and combined motion elements.
- *Other*: This set of parameters provide for the exchange and summation of system attributes such as cost and mass. In addition to this, a number of additional reserved data addresses ensures the flexibility of such a scheme by enabling user-defined parameters.

Intrinsic attribute communication

Intrinsic attributes are specific to a particular component. It is therefore, only necessary for these attributes to be available at design time for the considered component. These requirements can be met by the insertion of a dynamic data field for each component. It is proposed that this data field would provide constants and initialisation data for each model or electronic representation. Examples of such constants include the form factors for cast or milled gears (BS 436-1, 1967) or the properties for a shaft or coupling (BS 3170, 1972).

4.3 System resolution

As has been previously mentioned, the order of information exchange (data propagation) in a mechanical system is far more complex than in a fluid power system or an electrical circuit. This complexity arises as a consequence of the information dependency between elements and groups of elements. An example of this, is in the case of principal elements (core components such a shaft). Principal elements require that all related elements are resolved prior to their resolution in order that all inputs and outputs are considered.

To achieve the desired level of flexibility in the modelling approach, a strategy for system resolution is proposed which does not distinguish directly between assemblies and subassemblies, the latter being defined by a core object as in more traditional decomposition strategies (Culley & Theobald, 1997). The proposed scheme builds on the system representation discussed in section 4.1. This representation evaluates connections which can be used to classify system elements as either *unitary* elements (system inputs and outputs), *principal* elements (core components) and *binary* elements which convey inputs and outputs or connect other core components. The application of such a scheme to a machine system model is facilitated by the fact that core components are physically linked by the same components as those which transmit or convey system inputs and outputs.

In order to satisfy this dependency, the order of resolution must commence from the system inputs and propagate through connecting components until a core component is reached. Once each sequence of components that provides an input to the system has been resolved then representations that govern the core components may be interrogated. Following which, the sequences of components and their associated representations, which convey system outputs may be resolved. The order of system resolution is depicted in figure 4.10. This overall process is termed the *resolution episode*. The *resolution episode* commences at the unitary elements and propagates through the binary elements to the principal elements. Following which, the second phase of the episode initiates from the principal elements through the binary elements to the unitary elements or another principal element. The latter condition is necessary for resolving chains of components that link assemblies. This process is repeated until all principal elements are resolved. Because the parameters conveyed by the sequences of elements that connect principal elements (core components) are only available after the first resolution of principal elements a second phase of resolution is implemented. This second phase is also necessary for data arbitration and is particularly important where discrete components sizes are incorporated into the system. This is dealt with in some detail in chapter 5.

4.4 An integrated modelling approach

The fundamental components of a strategy for modelling mechanical systems and in particular their performance capabilities are; system representation, representing interactions and system resolution, described in sections 4.1, 4.2 and 4.3 respectively. These components provide the basis of what can be thought of as an integrated modelling approach. This approach addresses the deficiencies highlighted in chapter 3. The key functions of an integrated modelling approach for mechanical systems are:

- The ability to represent the performance of a system as a whole.
- The ability to construct a system from component based representations.
- Provide a neutral mechanism for the integration of third party component based representations.

- Determine a system of 'real' components, where these real components may be procured exactly as specified, and include both standard selected and standard designed components.
- Ensure that the system of components determined is physically realisable[2].

4.5 Concluding remarks

One of the objectives of an integrated modelling environment is to provide a systems approach that does not prescribe a particular form of representation or abstraction for individual components. To achieve this, the approach aims to interface the various current and emerging technologies for electronic representations. The realisation of these objectives are very important research issues identified by Subrahmanian *et al* (1993) and Culley (1999) respectively. In order to realise these goals, the approach of the work is to develop a neutral interface between component representations and the modelling environment. To achieve this, a protocol for communication in a modelling infrastructure has been generated which enables an assembly of engineering components to be considered in an integrated manner at the early stages of the design process. The problems associated with component interdependency and varying information requirements in mechanical systems is an important consideration. This complexity of information exchange between physically (i.e. directly) linked or connected elements in a system modelling environment is addressed by a communication protocol and a strategy for system resolution. This protocol enables information between connected components to be exchanged in order for the successful interrogation or execution of electronic representations. This is a particularly difficult task, which is frustrated by the large number of attributes to be handled in a mechanical system particularly when compared to fluid power or electrical systems.

The development of an overall attribute classification for mechanical components and the decomposition of these attributes to a range of parameters ensures that sufficient information is exchanged between connected components. The sufficiency of information is determined on the basis that both the accuracy and completeness of the range of parameters enable the generation of component attributes necessary for the execution of selection algorithms, catalogue searching or component design procedures from a variety of sources at the early stage of any design process.

It has been established that the ability to design a complete system within one environment is important for today's designers. Furthermore, domain specificity has been identified as a serious

---

[2] A physically realisable system is one which is capable of being fully assembled and functional from the range of components specified.

limitation of many modelling approaches for mechanical systems. Consequently, the systems modelling approach developed in this work does not distinguish between assemblies and subassemblies. The proposed modelling approach is equally applicable to systems comprising a single assembly or multiple assemblies. In addition to this, the ability to consider a complete assembly whilst utilising the features and functionality of third party representations is enabled by an architecture for interprocess communication and control of software applications, the development of which is described in chapter 7.

The strategy developed in this work, enables a system of components to be considered as a whole. That is to say the dependencies and interactions between related components can be represented. Furthermore, the approach provides for a component based modelling tool which is neutral with respect to the types of representation that may be incorporated. However, when considered collectively these component based models do not provide for the *handling* and *resolution* of conflicting component attributes or requirements. Consequently, if a feasible set of component specifications is to be determined these issues must be handled within the modelling approach. Therefore, two outstanding issues must be addressed; the development of procedures for arbitration and resolution of conflicts and the evaluation and assessment of component compatibility. The development of strategies that deal with these issues are discussed in chapters 5 and 6 respectively.

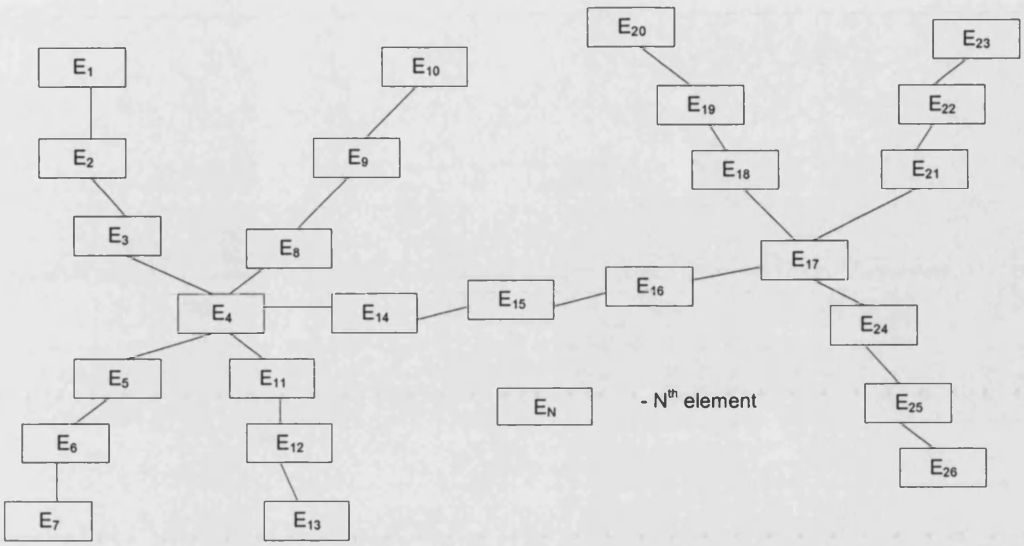


Figure 4.1 – An example schematic of an arbitrary mechanical system

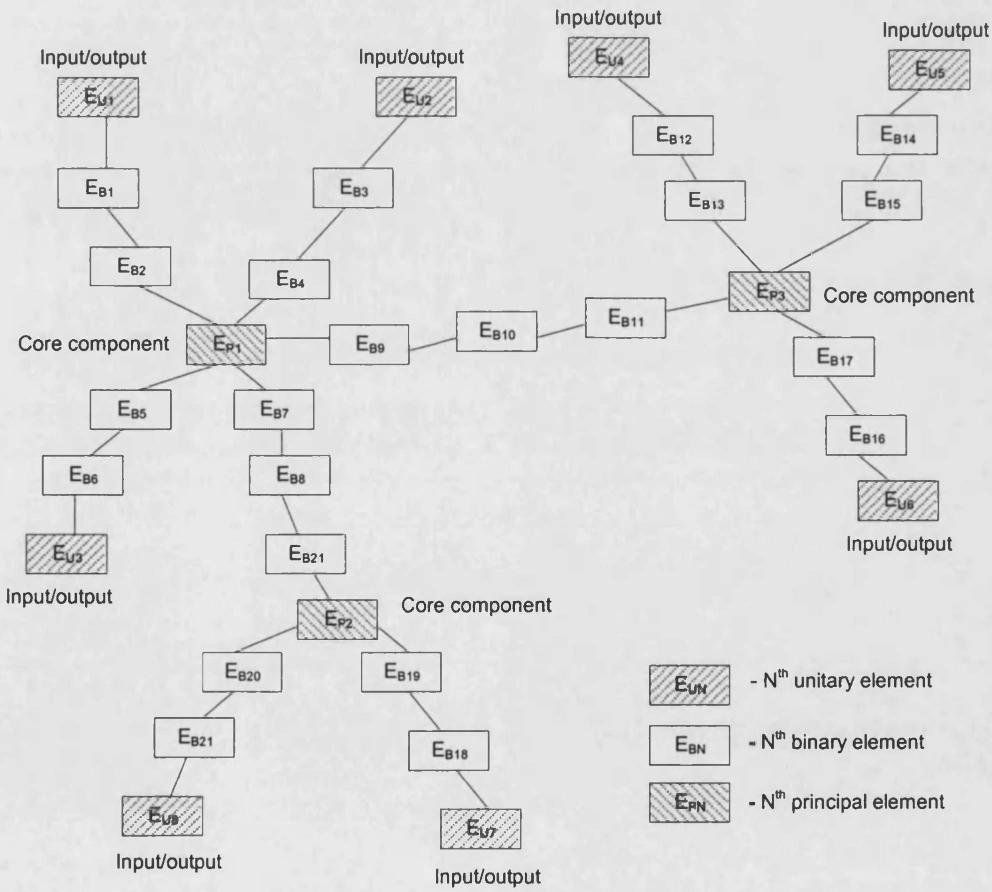


Figure 4.2 – An example elemental classification for an arbitrary mechanical system

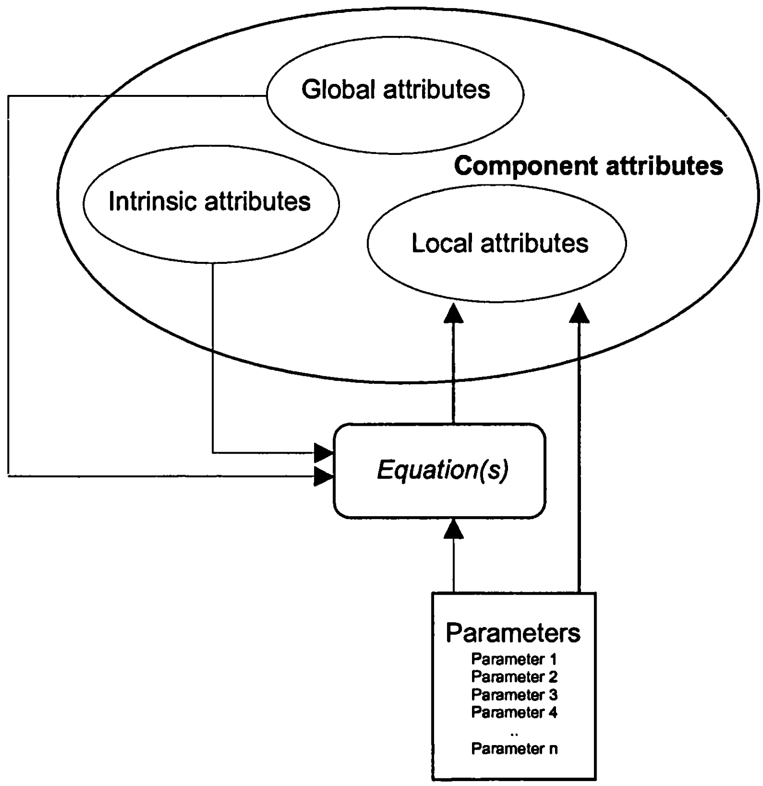


Figure 4.3 – The relationship between attributes and parameters

Shaft coupling	
Component attribute	**Attribute classification**
Bore1∅ - shaft input	Local
Bore2∅ - shaft output	Local
Diameter of coupling	Intrinsic
Length of coupling	Intrinsic
Hub ∅ of coupling	Intrinsic
Hub length	Intrinsic
Shaft insertion length	Intrinsic
Fastening screw size	Intrinsic
Peak torque	Local
Nominal torque	Intrinsic
Torsional stiffness	Intrinsic
Torsional deflection	Intrinsic
Max end loading	Local
Moment of inertia	Intrinsic
Mass	Intrinsic
Cost	Intrinsic
Shaft offset @speed	Intrinsic

Gear pump	
Component attribute	**Attribute classification**
Fluid type	Global
Fluid temperature	Global
Viscosity range	Global
Max contamination	Global
Drive type	Intrinsic
Operating pressure range	Intrinsic
Max pressure	Local
Max speed	Local
Min speed	Intrinsic
Rotation (clockwise/anti-clockwise)	Intrinsic
Torque	Local
Power	Local
Nominal volumetric	Intrinsic
Volumetric flow rate	Local
Volumetric efficiency	Intrinsic
Thread ports	Intrinsic
Inlet port size	Intrinsic
Leading dimensions	Intrinsic
Seals	Intrinsic
Mounting	Intrinsic
Mass	Intrinsic
Cost	Intrinsic

Figure 4.4 – Attribute classification for a shaft coupling and a gear pump

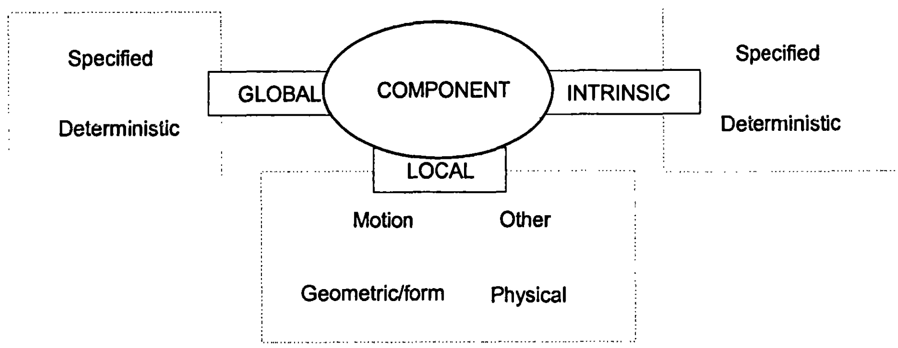


Figure 4.5 – Generic attribute classification structure

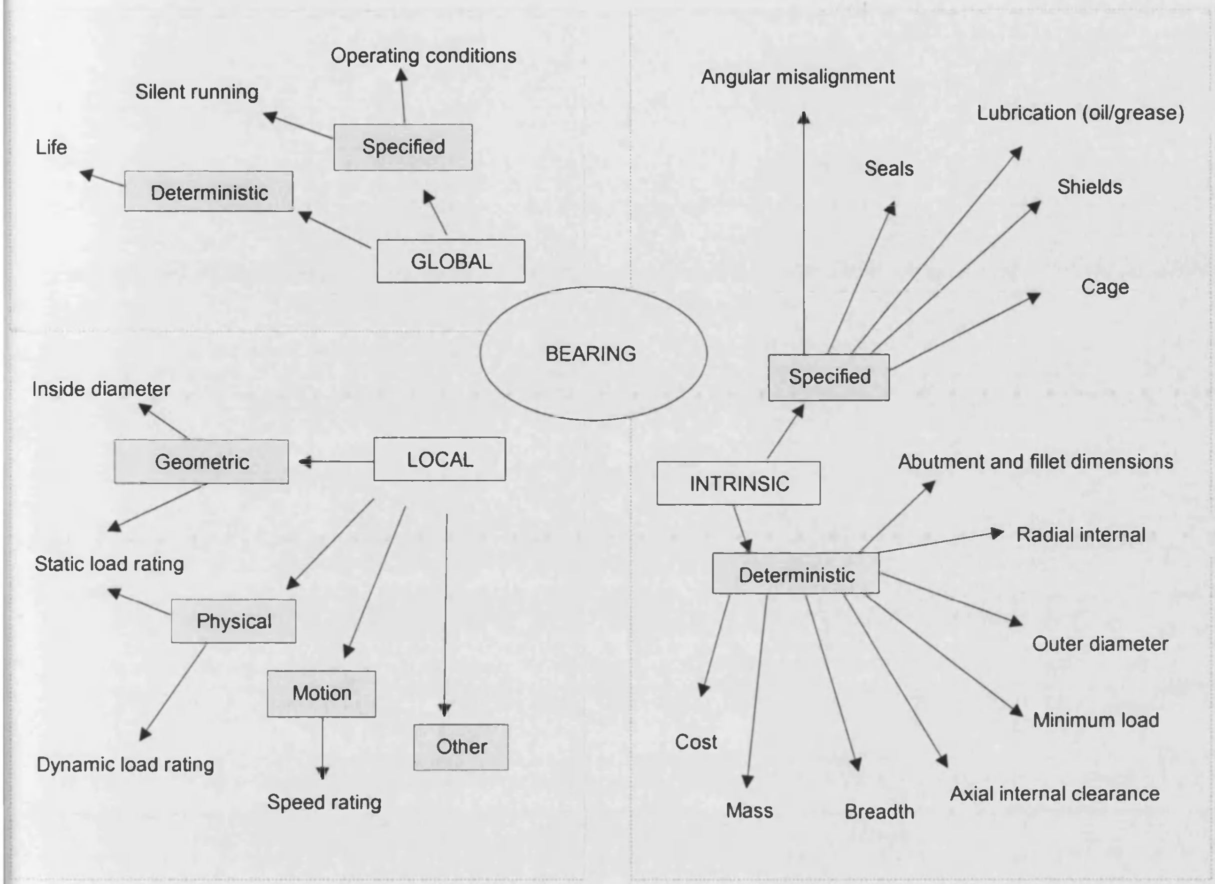


Figure 4.6 – Attribute classification for a cylindrical roller bearing

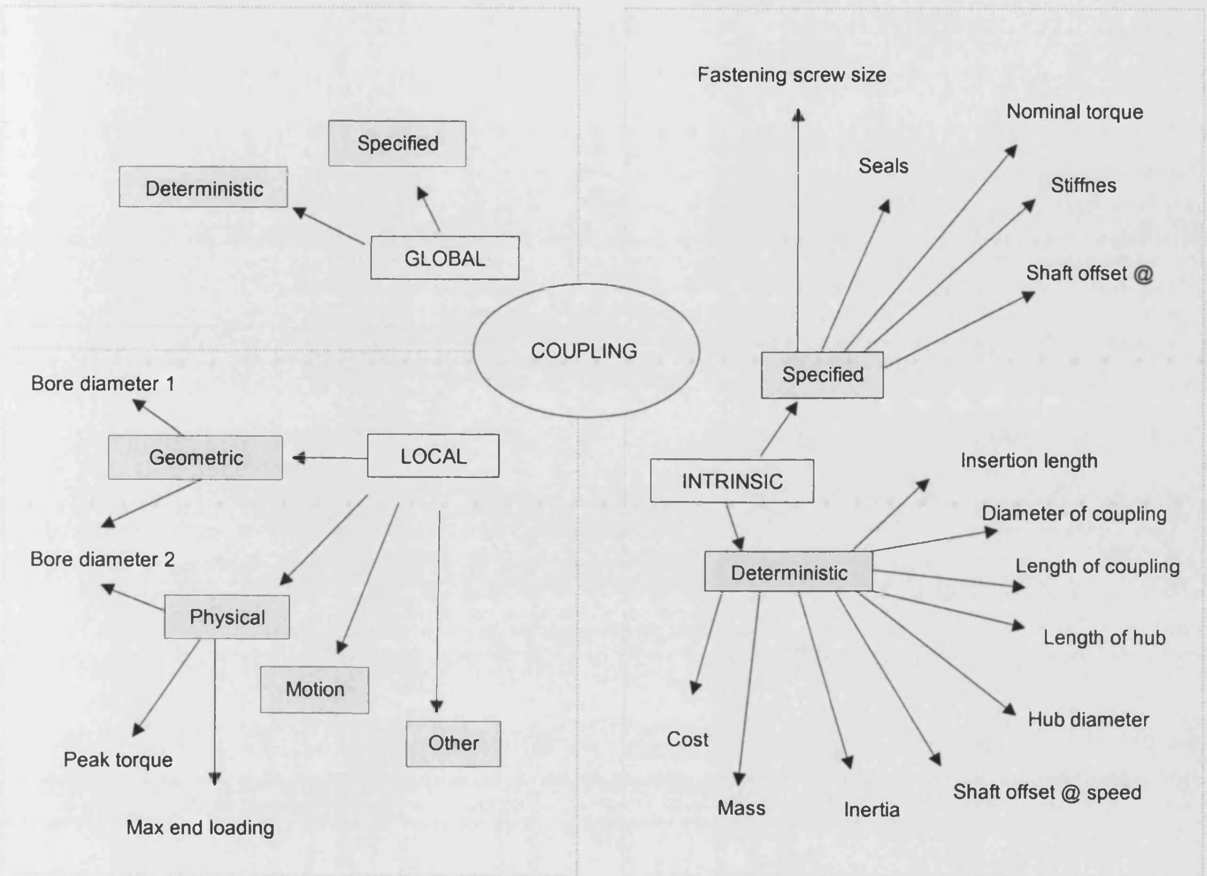


Figure 4.7 – Attribute classification for a coupling

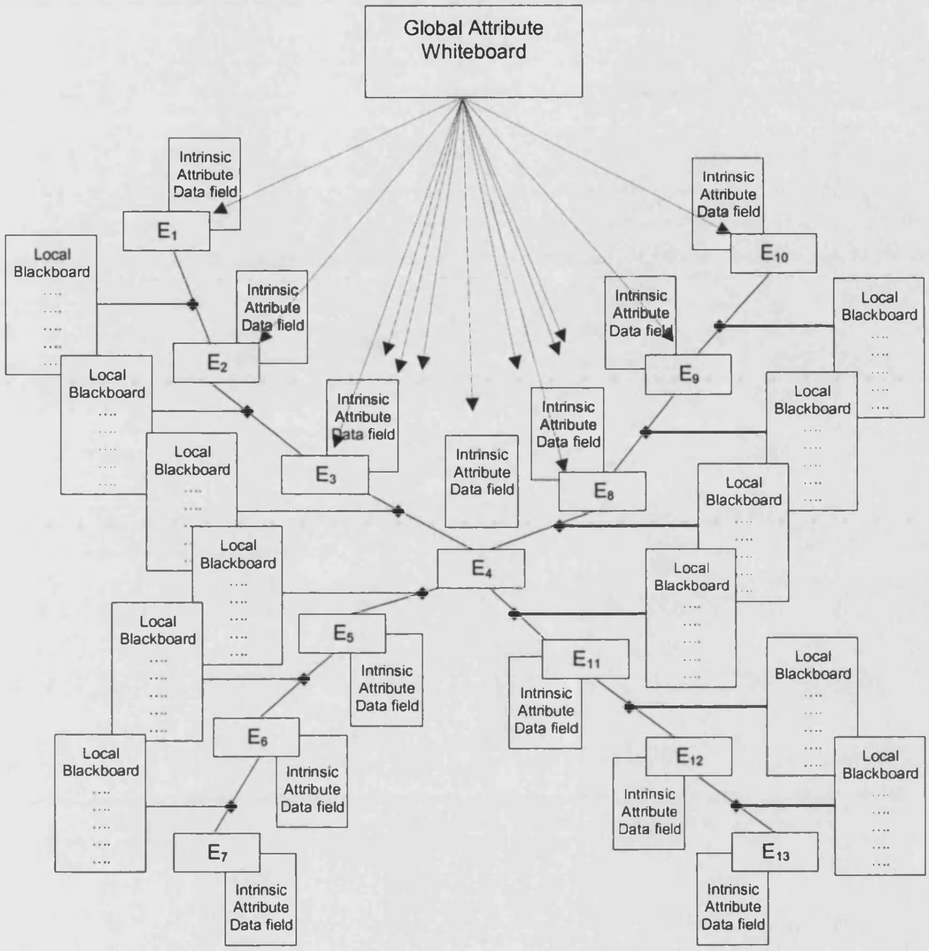


Figure 4.8 – The three mechanisms for data exchange in a system model

Whiteboard data packet	Whiteboard data fields (read status only)	Identifier Assembly No : Data Element No.
Global	Life (hrs)	WB1:1
	Reliability	WB1:2
	Safety factor	WB1:3
	Lubrication	WB1:4
	Temperature	WB1:5
	Humidity	WB1:6
	Other1	WB1:7
	Other2	WB1:8

Part (a) – An example 'global whiteboard' configuration

Blackboard data packet	Blackboard data fields (read & write status)	Identifier Board No : Data Element
Geometric/form	Geometric centroid (X)	BB1:1
	Geometric centroid (Y)	BB1:2
	Geometric centroid (Z)	BB1:3
	Max Dist form centroid (X)	BB1:4
	Max Dist form centroid (Y)	BB1:5
	Max Dist form centroid (Z)	BB1:6
Physical	Force (X)	BB1:7
	Force (Y)	BB1:8
	Force (Z)	BB1:9
	Torque (X)	BB1:10
	Torque (Y)	BB1:11
	Torque (Z)	BB1:12
Motion	Translation velocity (X)	BB1:13
	Translation velocity (Y)	BB1:14
	Translation velocity (Z)	BB1:15
	Rotational velocity (X)	BB1:16
	Rotational velocity (Y)	BB1:17
	Rotational velocity (Z)	BB1:18
Auxiliary	Cost (£)	BB1:19
	Mass (kg)	BB1:20
	Inertia (X)	BB1:21
	Inertia (Y)	BB1:22
	Inertia (Z)	BB1:23
	Other (1)	BB1:24
	Other (1)	BB1:25

#(Note: the Auxiliary data packet is structured to be flexible, the elements in the above example are included are to demonstrate possible alternatives)

Part (b) – An example 'local blackboard' configuration

Figure 4.9 – Extents of data exchange in the modelling approach

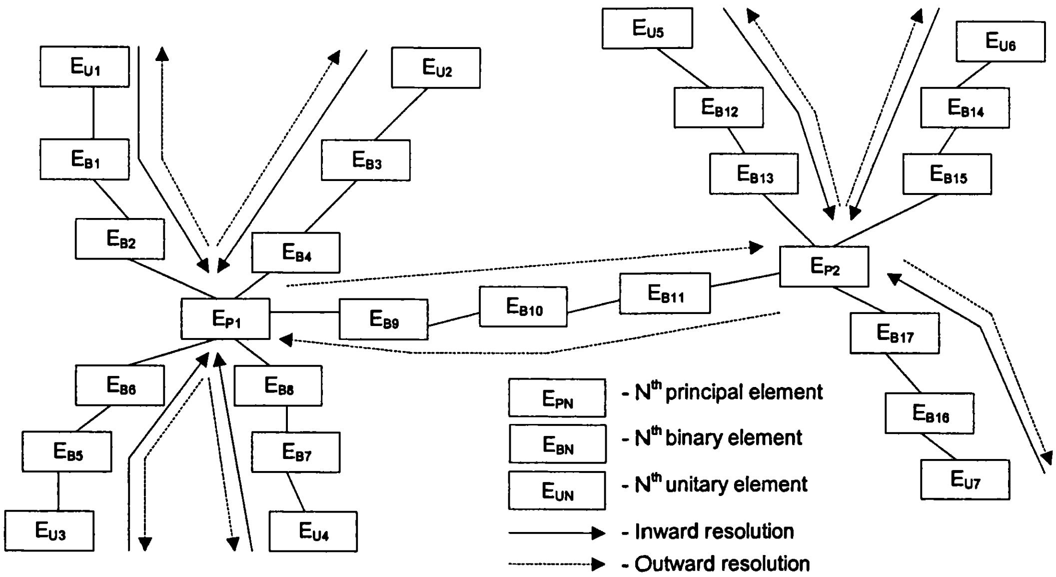


Figure 4.10 – Order of system resolution in the modelling approach

Chapter 5

Data arbitration and conflict resolution

Chapter 4 discusses the development of a modelling approach for mechanical systems and in section 4.4 the key functions of an integrated modelling environment are described. The creation of an integrated modelling environment addresses the second hypothesis set out in this work. One of the key features of an integrated modelling environment is the ability to determine an overall system solution of real components that is physically realisable[1]. This chapter and the following chapter focus on the development of what can be thought of as the support functions necessary to ensure that a feasible system of components is determined during the modelling episode. In order to generate a feasible solution two criterion must be satisfied:

- The embodied solution must be free from conflicts and ambiguities.
- Mechanically coupled components must be compatible.

This chapter addresses the first of these issues, through the development of a strategy for data arbitration within the systems modelling approach.

5.1 Data arbitration in a system modelling environment

Within any complex system containing a large number of elements there is a need for data arbitration or conflict resolution, especially when the system is largely nonholonomic (Flood & Carson, 1988). A system which is nonholonomic contains elements or parts that are temporarily outside the system control. This is certainly the case where third party component representations, which are largely independent, need to be integrated in a systems approach. For these *component based* representations to be integrated, a complete set of performance attributes for each component needs to be held within the system. In addition to this, a set of design constraints or desired parameters for the considered component must also to be generated.

---

[1] A physically realisable system is one which is capable of being fully assembled and functional from the range of components specified.

In the systems modelling approach developed, the attributes for each component are contained within a data field, whilst the design constraints may be considered to be those parameters that are acquired through relations with physically connected components. From these two data sets a master set needs to be generated which forms the basis of the data for the interrogation or execution of component based representation. This data set comprises the desired performance and physical requirements that will be used by the representation to determine a suitable component. It is during the generation of this data set that conflicts may arise between design constraints and the desired performance attributes for the component. These conflicts must be resolved and a single attribute value determined if the modelling episode is to be successful. An example of a conflict is between the diameter of a shaft and the internal diameter of a bearing. The component based representation governing the selection of a bearing will determine a bearing with a given internal diameter, from the discrete range available. This parameter will then be exchanged with the shaft model. If the values are not equal, say for example that the designer has specified a desired value for the shaft diameter, then a decision must be taken to determine which value to maintain.

5.2 Nature of conflicts in an integrated modelling environment

For the modelling approach developed in this work, critical conflicts arise between communicated *parameters*, discussed in chapter 4, *constraints* and *primary component attributes*. These *primary component attributes* are the fundamental attributes upon which a component is specified or selected and may include both physical attributes and performance attributes. The identification of these primary attributes is dependent upon the level of dependency the particular component has with other coupled elements. An overview of primary and secondary attributes for a bearing and a shaft is provided in figure 5.1. The *secondary attributes* of a component may well provide constraints on performance, but are more often than not either fixed values or follow a predetermined discrete range for the considered component, such as the permissible angular misalignment of a bearing. Consequently, mechanisms for arbitration only need to be implemented for this primary set of attributes. To address the constraints imposed by secondary attributes a method for *constraint specification and evaluation* combined with a strategy for *bounds evaluation* are described in section 5.4. Conflicts between communicated parameters and component attributes arise because of the various methods of formulation for attribute values. In the modelling approach there are five mechanisms for the generation of attribute values:

- *Initialisation values* - In order to construct a system model each component possesses a data field comprising initialisation values for all its attributes. This enables the execution of the appropriate electronic representation with the minimum level of attribute specification by the designer.

- *Specified values (component level)* - The value of attributes in the component data fields can be specified by the designer, and these desired values must be maintained by the modelling system.
- *Determined values* - The interrogation of electronic representations during model resolution generates either a partially or completely new set of attribute values for a component.
- *Discrete values* - Standard selected components and many standard designed components possess attributes which follow a finite predetermined range. Such discretised attribute values need to be considered in the modelling approach.
- *Specified values (system level)* - Global attribute values are set by the designer at a system level or an assembly level and may well differ from that specified in component data fields.

During system resolution, related attributes may have been formulated by any one of these mechanisms, and it is this method of formulation or mechanism that provides the basis for arbitration within the modelling approach. The arbitration of attributes must therefore be undertaken through consideration of these five factors.

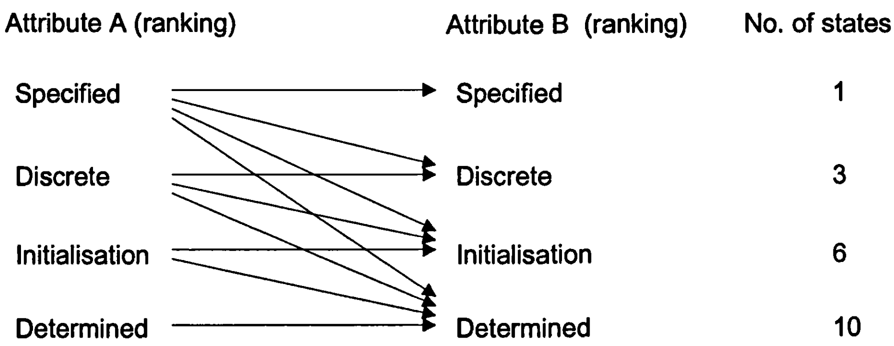
5.3 An agent based approach to conflict resolution

In collaborative design a conflict arises when there is a decision situation where two or more agents with individual preferences disagree on a mutually shared design goal (Sreeram, 2000). The requirements for conflict resolution in the proposed modelling approach can be considered to be an agent based problem as a number of key characteristics are necessary for system resolution. These are defined by Sreeram (2000) as:

- Autonomy: the ability to make decisions independently.
- Ability to communicate: the ability to communicate with a formal language.
- Ability to resolve: the ability to resolve conflicts by means of some formalised mechanism. This could be domain specific rule-based theory or game theory.
- Interoperability: the ability to interact with other agents from different software environments or platforms.
- Pro-activeness: the ability to take the initiative in complex problem solving environments.

In the context of this work, agents may be considered to be individual electronic representations. These are independent software objects that determine a feasible component specification from a set of performance requirements. Communication and interaction between these agents is enabled by the modelling approach. The requirements for autonomy and conflict resolution cannot be met by existing third party representations. As a consequence, virtual agents are implemented. These *virtual agents* provide for the autonomy, communication, resolution and interoperability outlined above. Virtual agents are secondary software objects that act as intermediaries between the component based representation and the system modelling environment. An overview of the architecture for a virtual agent is depicted in figure 5.2. In the approach developed, the virtual agent is an independent software module that is associated with a particular component representation. Attributes for the considered component and parameters from related components are parsed to the agent. These values are then arbitrated by the agent and a revised set of performance attributes are compiled for the interrogation of the component representation.

The virtual agents utilise a generic strategy based on a number of rules and precedence's to guide the agent. For this work, conflicts occur between attributes and their numeric values. This type of conflict can be dealt with by a utility based approach. The suitability of a utility based approach for the resolution of parametric conflicts is discussed by Sreeram (2000). The approach involves the assignment of indexing or rankings to conflicting variables. For the five types of conflict identified in section 5.2 four classes of attribute can be defined. These are *initialisation*, *specified*, *discrete* and *determined*, and depend on the method of formulation of the attribute value. These four classes give rise to ten potential states of conflict between the rankings of two different attribute values:



In the case of four out of the possible ten states of conflict the conflicting parameters are defined through processes of equal status, i.e. discrete and discrete or determined and determined. For all conflict states where rankings are equal, with the exception of initialisation attributes, hard[2] conflicts are generated (Klien, 1990). The resolution of this type of conflict is frustrated by the fact that often a compromise is not obvious and may require detailed negotiation or an understanding of other system elements to determine an acceptable compromise. Consequently, these can rarely be dealt with in a deterministic manner and are not addressed in this work. Rather the designer is notified of the conflict and prompted to take action. In order to enable the resolution of the six remaining states of conflict, a ranking strategy is developed which corresponds to the method of formulation of the attribute.

- *Specified attributes* or *parameters* are derived from component attributes for which the designer has specified a desired value and hence carry the highest precedence, with a numeric value of 3.
- *Discrete attributes* or *parameters* are those values derived from attributes provided by a component based representation and that follow a discrete finite range of values. This process of attribute definition possesses the next highest precedence with a value of 2. This is because it is desirable to maintain a discrete value over a continuous value providing that the required performance can still be delivered.
- *Determined attributes* or *parameters* are those values derived from attributes provided by an component based representation, and that follow a continuous range. These possess a ranking with a numeric value of 1.
- *Initialisation attributes* or *parameters* provide the initialisation data for the component based representation and carry the lowest precedence with a value of 0.

During the modelling episode, virtual agents compare the rankings of component attributes in the data field against those of communicated parameters from connected components. If both the communicated parameter and desired attribute possess a ranking of equal status then the conflict is registered in the system log and the designer must undertake manual arbitration to determine a comprise. For conflicts involving rankings of unequal status the conflict can be automatically resolved. The generic process for this resolution procedure is depicted in figure 5.3.

[2] Hard conflicts are the most difficult class of conflict to resolve as often a straightforward win or lose situation is not possible.

Communicated parameters and associated rankings are evaluated by the virtual agents against corresponding attributes and rankings from the component data field. The arbitrated value and associated ranking are maintained by the system. This ensures that more important values are preserved by either overwriting or not propagating less important parameter and attribute values, and that high ranking design constraints are propagated through the model. The arbitrated parameter values are passed to the component representation whilst the arbitrated flags are passed back to the local blackboard[3] and the component data field respectively. The component representation is then interrogated and the resulting attribute values are passed to the data field, and parameters to the local blackboard associated with the next component to be resolved. In this manner, design constraints are propagated throughout the system model.

5.4 Constraint specification and boundary analysis

As previously discussed in section 5.1, agent based arbitration is not essential for all classes of conflict. For conflicts where secondary attributes are involved, the identification and notification of a conflict is often sufficient. Secondary attributes typically describe intrinsic component values and are not directly related to the physical connections with other components. Secondary attributes may include properties such as lubrication type or angular misalignment. Unlike primary attributes which are critical to the success of system resolution, secondary attributes can be resolved post-system resolution. In order to represent secondary constraints between elements, logical conditions are constructed between the attributes of related components. These constraints are evaluated concurrently during resolution. An overview of this approach is depicted in figure 5.4. These conditions or constraints are generally expressed as numeric equalities, although string comparisons can be evaluated, but this is restricted to exact matching only. This feature can be utilised for ensuring attributes such as lubrication type are consistent throughout a design. If the constraints are not satisfied the designer is notified and prompted to take action in order to resolve the conflict and satisfy the constraint.

Further to constraint evaluation and data arbitration, there is also a need for bounds evaluation. This is particularly important where component attributes are limited or are only available over a predetermined range, which is often the case with standard components. If these limits are exceeded by performance requirements then it may be necessary to change the component type or

[3] Local blackboards are the transfer mechanism for the exchange of design parameters and are discussed in chapter 4.

alter the configuration. If the component cannot feasibly be procured and the system is resolved then the solution may well have to be reconfigured later, especially if it has been optimised. It is therefore necessary to determine whether bounds have been exceeded prior to optimisation. The procedure for boundary analysis is shown in figure 5.5. The process is invoked after data arbitration and evaluates any bounded performance variables. In this manner, the designer can be notified immediately if the desired attribute values for a component exceed the performance envelope of the component or available range of the supplier or manufacturer.

5.5 Concluding remarks

Within a complex system, the arbitration of related attributes between coupled elements is essential so that conflicting requirements can be resolved and a system solution determined. This chapter describes two levels of conflict resolution within a mechanical system modelling approach. The first of these is at a system level. Constraints between elements are specified as logical conditions, which may be evaluated concurrently during resolution. These conditions are generally expressed as numeric equalities, although string comparisons can be evaluated, but this is restricted to exact matching only. This feature can be utilised for ensuring attributes such as lubrication types are consistent throughout a mechanical system. If the constraints are not satisfied the designer is notified and prompted to take the appropriate action to resolve the conflict and satisfy the constraint. The second level of conflict resolution is at an elemental level and can be considered to be agent based. This agent based data arbitration is performed by virtual agents. These virtual agents are secondary software modules associated with each component based representation. This is necessary because existing representations do not provide the functions necessary to undertake decision making independently. These agents ensure that critical conflicts are resolved between connected components and that important design constraints are propagated through the system. A generic strategy for agent based conflict resolution, which compliments the integrated modelling approach developed in this work, is discussed and incorporated into modelling approach.

	Bounding			Parametric status			Arbitration		
Attribute	Min	Max	Discrete	Driven	Constant	Driver	Primary	Secondary	Bounds check
Power						✓	✓		
Node diameter	15†	420†				✓	✓		✓
Node length	11†					✓	✓		✓
Speed						✓	✓		
Material stress						✓		✓	✓
Safety						✓		✓	✓
Mass				✓					
Cost				✓					

† Limits taken from the bearing catalogue

Part (a) – Attribute classification for a parametric model of a shaft

	Bounding			Parametric status			Arbitration		
Attribute	Min	Max	Discrete	Driven	Constant	Driver	Primary	Secondary	Boundary check
Internal diameter	15	420	✓			✓	✓		✓
External diameter	35	620	✓			✓	✓		
Breadth	11	170	✓			✓	✓		✓
Dynamic load	12500	2920000	✓	✓			✓		
Static load	10200	4900000	✓	✓			✓		
Speed	950	22000	✓			✓	✓		✓
Life					✓	✓			
Lubrication					✓				

Part (b) – Attribute classification for an electronic bearing catalogue

Figure 5.1 – Example attribute classifications for electronic component representations

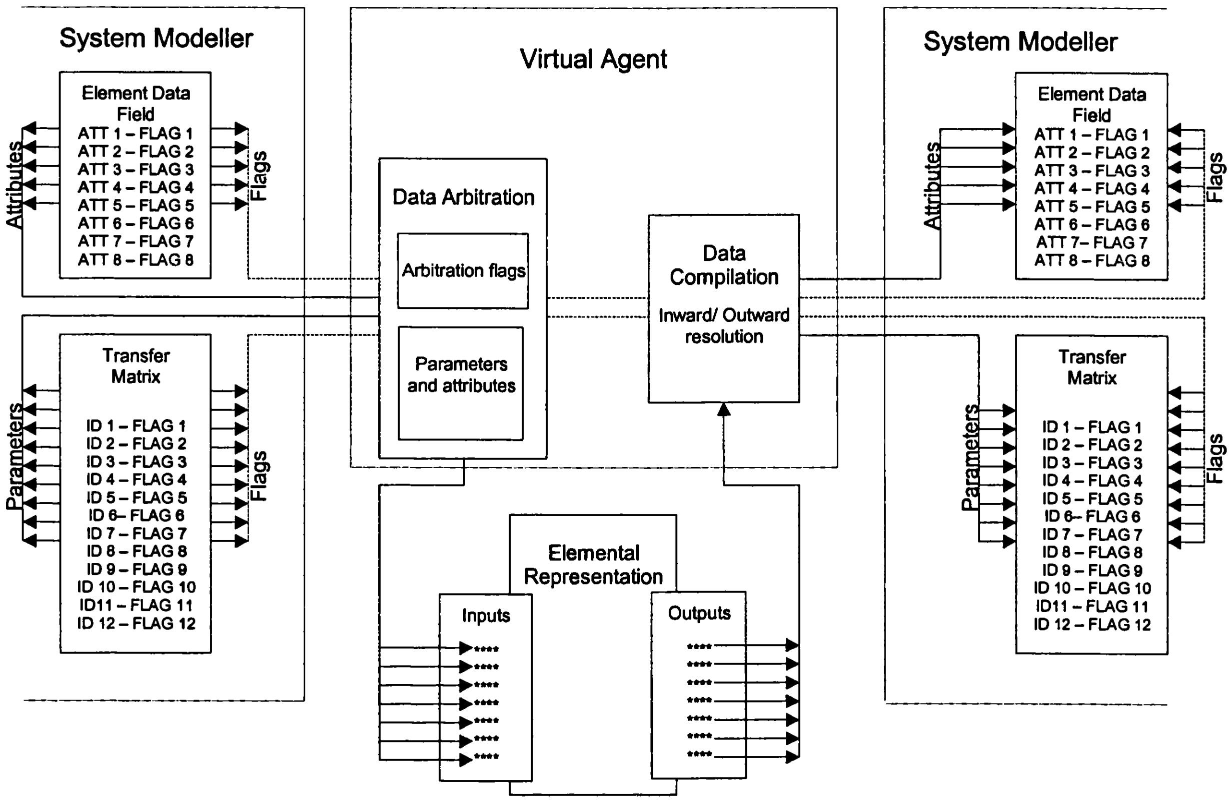


Figure 5.2 – The function of a virtual agent in the integrated modelling approach

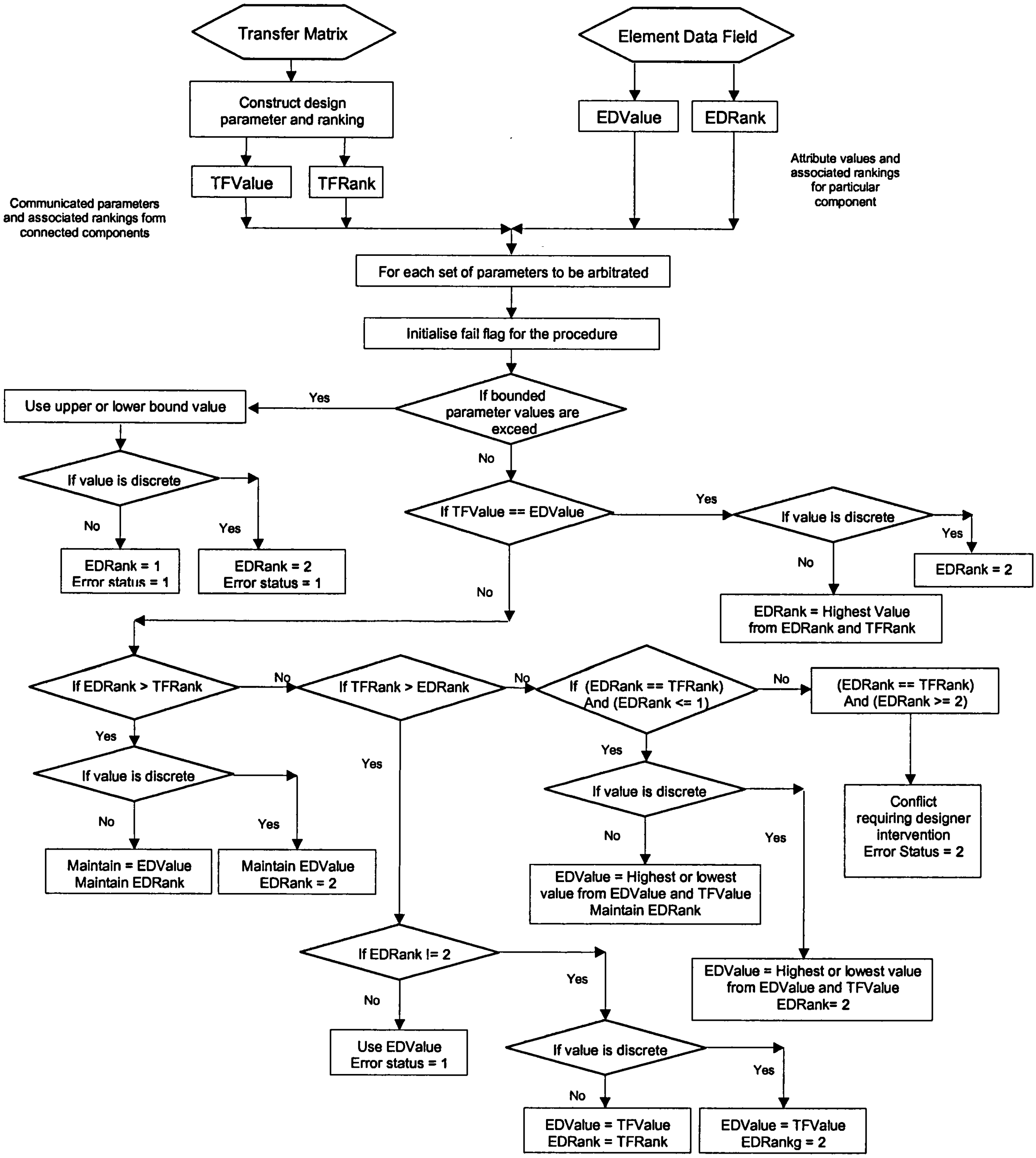


Figure 5.3 – The generic structure for data input, arbitration, assimilation and output in component based models

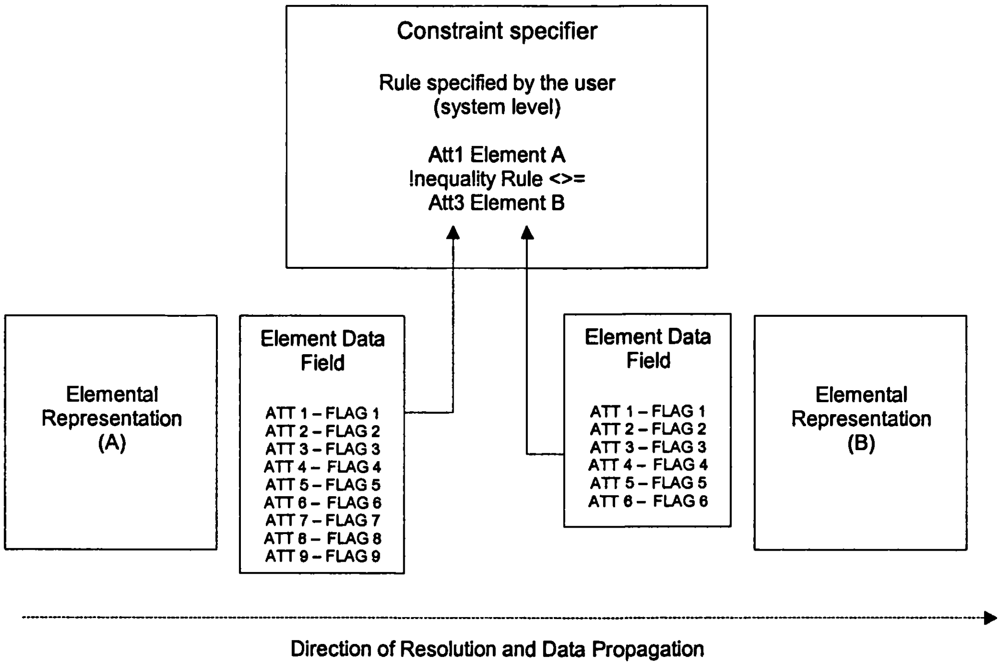


Figure 5.4 – Constraint evaluation within an integrated modelling environment

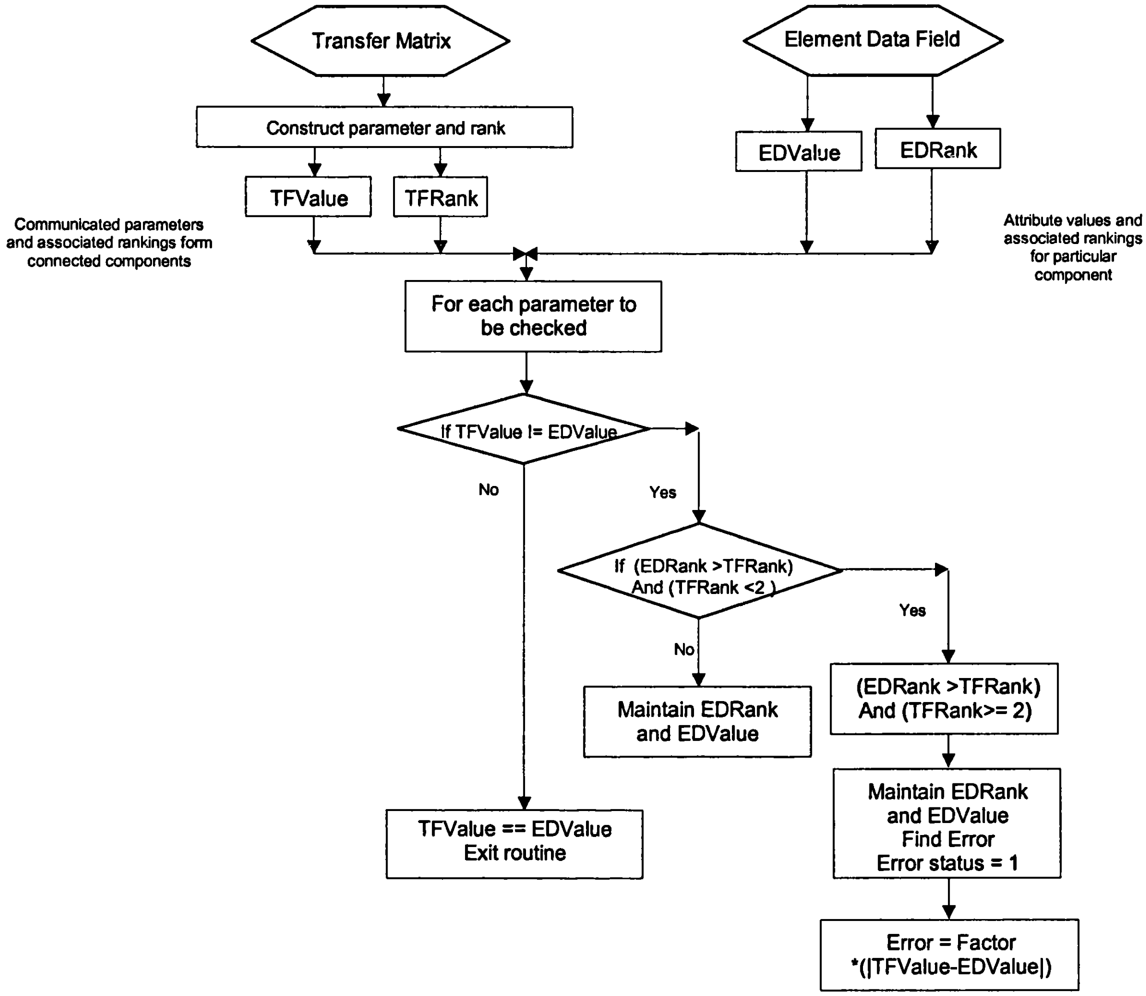


Figure 5.5 – A generic structure for bounds evaluation in a virtual agent

Chapter 6

Compatibility analysis

Chapter 5 introduces two criterion that must be satisfied for the determination of a physically realisable system. These are that the embodied solution must be free from conflicts and mechanically coupled components must be compatible. The issue of compatibility analysis is developed in this chapter and a strategy to assist the identification and utilisation of compatible components in an integrated modelling approach is developed.

The ability of standard components to deliver increased quality and considerable time savings for the designer has been discussed in earlier chapters. In addition to the advantages previously outlined, improved economy and performance can be obtained by the effective utilisation of various configurations of standard components. In order to establish the most effective or optimum utilisation, the designer must evaluate various component types, sizes and combinations. A *component type* is considered to be a specific range of mechanical elements from a particular manufacturer, whilst *component sizes* relate to the range of discrete element sizes available for a particular *component type*. During this evaluation, the designer must consider performance and a variety of aspects such as reliability, suppliers, cost, maintenance and internal practices. The pressure for reduced time to market does not leave room for time-consuming trial and error approaches. This can be alleviated by computer modelling and in particular systems modelling. However, if these approaches are to be truly useful they must provide for the capability to assess the compatibility of connected components both quantitatively and qualitatively. The former pertains to whether the components are physically connectible and matched in terms of their performance capabilities, whilst the latter relates to the utilisation of the components, with respect to best or recommended practices. These may include preferences for suppliers, cost considerations, reliability issues, component standardisation, scalability and rationalisation of designs within the company. These are not meant to be exhaustive but merely illustrate some of the qualitative considerations which a designer must undertake.

A support tool to aid the embodiment of systems, however defined, needs to address a number of issues, and in particular the ability to consider many more alternatives within shortened activity times and the ability to evaluate achievable component configurations. To address these issues, provision for the identification and rejection of incompatible component types as well as the suggestion of alternative component types must be incorporated. In addition to this, the recognition of acceptable or unacceptable component combinations or chains of components is demanded. For example, a novice designer may attempt to connect incompatible components together, whilst a more experienced designer may utilise chains of components that are compatible but not practiced for reasons such as reliability, stiffness or cost. Therefore, designers need to know whether elements are compatible and whether the combination of elements they have chosen are best-practiced or acceptable alternatives. For the purpose of this work, the term best practiced may relate to either internal or corporate practices as well as accepted or documented design practices. For the task of identifying compatible elements, various levels of compatibility analysis have been practiced for some time in both electrical and fluid power design tools. These techniques are discussed in later sections with respect to the generation of a similar supportive approach for mechanical systems. Firstly an overview of selection issues for mechanical systems is provided.

6.1 Selection issues for systems of mechanical components

For today's engineer the selection of standard components involves the consideration of many more factors than solely; "does this component deliver the required performance?" Because of global markets and global distribution networks, combined with the vast number of different suppliers and component types, the design team must now consider issues covering not only performance but also aspects such as suppliers, cost, delivery, reliability, compatibility and managed services to name but a few. Figure 6.1 illustrates the key activities identified in this work and depicts the individual tasks for the design of a mechanical system with standard components. These activities are discussed in the following sections and are categorised from studies into machine systems design. The activities can be separated into four phases; *identification and selection of components*, *component matching and compatibility analysis*, *optimisation of sizes and configuration*, *and detailed design*. These phases are separated in the process overview in figure 6.1, however, there is a high level of dependency and recursion between each phase and their included activities. The core tasks or activities in each of the phases are now described and research methods which support each phase are discussed.

6.1.1 Identification and selection of components

During the first phase of the proposed process the designer is concerned with identifying suitable component types and associated manufacturers. To undertake this, a number of authors discuss methods for the mapping of function to types of mechanical component (Counsell *et al*, 1999; Hundal & Langholtz, 1992), termed functional decomposition, this approach is also a key feature of the TRIZ methodology (TRIZ Journal, 2001). For the identification of preferred suppliers or particular manufacturers, Ellram (1990) discusses methods for the evaluation of relationships with suppliers, whilst Culley and Allen (1999) describe an approach to capture informal information regarding the effectiveness and performance of suppliers and particular component types.

6.1.2 Component matching and compatibility analysis

The second phase of the process involves many different tasks, which are highly dependent on each other and must be considered concurrently in order for the successful completion of this phase. This dependency is represented in figure 6.1 by the interconnections between each task. These individual tasks can be further grouped into four activities: *system topology or layout*, *system performance*, *connectivity and performance matching*, and *complementary assessment of coupled components*, depicted by the hatched regions in figure 6.1. For the first of these activities there are many tools which consider geometric relations between parts, these include the CAD packages; WAVE (part of Unigraphics UGS, 2001) and Pro Engineer (PTC Inc, 2000), both of which possess parametric modelling capabilities for assemblies.

To support the undertaking of the second activity, a number of approaches for performance modelling and simulation are becoming available, which enable the modelling and analysis of systems based on standardised components or models (AMESim, 2001; MSC Software, 2001). These approaches support the conceptual and embodiment phases of the traditional design process and assist the designer in rapidly embodying a design solution. This embodiment is frequently computer based and attempts to populate the conceptual configuration with a set of mechanical components that meet the desired overall performance characteristics. The automation of the embodiment process enables many more component configurations and combinations to be evaluated by the designer in a relatively short time period

For the third and fourth tasks in this phase of the proposed process; performance matching, connectivity and compatibility analysis there is little documented work, where systems of standard components are considered. To summarise these tasks, their primary function is to ensure that geometric interfaces are compatible (for example, this may include evaluation of fits and the ability to accommodate deflections), the magnitudes of energy transfer are acceptable, and that where possible preferred component types and combination are used. Current practice requires the designer to perform such evaluations by hand, which is inherently time consuming and frustrates the ability to determine an optimum solution. The development of methods which deal with this aspect of the embodiment process are therefore essential for the generation of the most effective overall design solution.

6.1.3 Optimisation of sizes and configuration

The third phase of the process involves the optimisation of a particular configuration of components, where the configuration represents the types of component, their arrangement and their preliminary sizes. The optimisation strategies applied to such problems typically vary component attributes, evaluating system properties after each iteration, in order to converge on a solution which best achieves the desired attributes for the considered system. The examples of optimisation goals detailed in the lower portion of figure 6.1 merely represent some typical objectives (Papalambros & Wilde, 1988; Adeli, 1994; Samuel & Weir, 1999). The actual goal function(s) must be generated by the designer and will depend on the original design specification. Extensive research has been undertaken into developing optimisation strategies for searching both continuous and discrete solution spaces (Walsh, 1975; Hajela, 1999; Sen & Yang, 1988), and although these have not been used extensively in these applications they are considered suitable for problems involving standard components which produce discrete solution spaces.

6.1.4 Detailed design

The final phase of the process includes many of the typical tasks for detailed design (Pahl & Beitz, 1996). These may include the specification of the intricacies for say, component housings, mountings or fixtures. This phase may also include design for X type activities (Ullman, 1992), although if different component sizes are considered the designer may need to re-enter phases two and three of the process. Traditional CAD systems (Autodesk Inc, 1997; SolidWorks Corporation, 1998; EDS Inc, 2000) have been developed for this phase of the design process. The majority of these systems enable the geometry of standard components in the form of two-dimensional and three-dimensional models to be imported from libraries, often supplied by manufacturers or included in electronic catalogues (Culley & Webber, 1992). These libraries also include mountings and standard fixings, removing much of the detailed work.

6.1.5 Overview

The previous sections discuss the individual phases of the proposed 'design process' for embodying mechanical systems with standard components. These sections also provide a brief overview of current technologies and software tools which support the various activities. This review highlights the fact that there is a lack of supportive methods for the second phase of the process and in particular what has been identified as compatibility analysis. Furthermore, the overall process depicted in figure 6.1, shows that there is a high dependency between each of the phases. This demands that the various activities are considered concurrently in order to undertake the process effectively. To address these important issues, the following sections discuss and develop the requirements for compatibility analysis in mechanical systems, and review similar approaches adopted in other engineering domains, namely fluid power and electrical circuit design. A strategy for compatibility analysis is developed which compliments the modelling approach created in this work. The incorporation of compatibility analysis into the modelling approach enables much of the embodiment process to be undertaken in a single environment. The collective consideration of system performance, geometry and component selection issues are essential for the effective and successful selection and specification of a system of standard components.

6.2 Review of compatibility analysis in engineering

Due to the desire to reduce development times and the commitment of final cost obligated at the conceptual phase of the design process, the ability to test for 'achievable designs' becomes increasingly more important. Consequently, the integration of compatibility analysis routines into software based design environments is beginning to evolve. Such environments range from the configuration of PC hardware (Coleman *et al*, 1996) to the compatibility of information systems (Mailoi *et al*, 1994). For all these various applications, compatible elements are defined as "*those elements which can be used together without modification or adaptation*" (The Concise Oxford Dictionary, 1990), and do not produce any physical effects that may adversely affect the performance of the system considered. In the domains of fluid power systems design and electrical circuit design compatibility analysis techniques have been successfully incorporated into a number of support tools. In the electrical domain compatibility analysis deals with electro-magnetic compatibility utilising complex numerical methods to model the behaviour of fields (Parry *et al*, 2000). In contrast to this, work in the design of fluid power systems focuses on the compatibility of component models, with respect to the matching of input and output parameters as well as the continuity of units of measurement (Sidders *et al*, 1996).

In machine systems design, compatible designs can be considered to be those that are 'achievable' and comprise 'preferred' or 'best practiced' configurations of components.

- *Achievable designs* in this context are those that are fundamentally based on existing technology and principles. Such designs incorporate standard components or standard designed components, and are configured so that components are physically connectible and mutually complement each other, in terms of the type(s) and magnitudes of their inputs and outputs. As the number of elements that violate these conditions increases, so the confidence in the achievability of the design reduces.

- *Preferred designs* utilise configurations and combinations of components that are deemed to be suitable or better performing for the particular application. Reasons for this 'perceived better performance' range from cost concessions provided by certain suppliers or a drive to use existing stocks to well tested and reliable component combinations, an important consideration in aeronautical disciplines (Keynote, 1998). As the number of preferred components reduce the subjective or qualitative attributes, such as reliability or efficiency, become less certain.

For the purpose of machine systems design, the objectives of compatibility analysis as defined above are concerned with the assessment of connected components rather than system effects, such as the fields produced by electrical circuits. Hence, an elemental approach is all that is required. This elemental approach evaluates connected components and sequences of coupled components throughout the system. This is similar to the approach adopted in the fluid power domain, where attributes between coupled components are evaluated. However, modelling environments in the fluid power domain mainly deal with flow rate, pressure and consistency of units. In order to develop an elemental approach for compatibility analysis of machine system models a number of issues must be addressed. These are discussed in the next section.

6.3 Compatibility analysis in mechanical systems

For the purpose of this work, compatible elements are defined as those which are capable of being used in combination and are 'well suited'. This latter aspect ensures that coupled components are matched in terms of their capabilities and do not impose any adverse effects on to each other. In order to evaluate the compatibility of a system of mechanical elements three activities are proposed; *connectivity analysis, performance matching* and *complementary assessment*, illustrated in figure 6.2.

- *Connectivity analysis* ensures that the geometric interfaces are matched and that energy interfaces are compatible. This ensures that coupled components fit together and that energy can be transmitted in the desired manner across their interface. An example of the importance of this latter condition is for the selection of a bearing on a shaft. If the shaft transmits an axial load and a radial load then, whilst many bearings may be geometrically compatible, only bearings which support both loading conditions are suitable for the particular design.

- *Performance matching* ensures that the magnitudes of energy transfer are acceptable output and input levels for coupled components. This matching may not be exact, the designer may wish to include safety factors or additional capacity for variations in performance conditions. In addition to this, performance matching encompasses component attributes that are not exchanged at the interface. These may include lubrication type, material properties, working temperature, life and reliability.
- *Complementary assessment* provides for the qualitative considerations which the designer must undertake. These may include cost, reliability, manufacturer and supplier issues and recommended or standard practices. The objective of complementary assessment is to identify particular components or combinations which are preferred. An example of complementary components is in the case of the attachment of a gear to a lay shaft. It may be that the manufacturer recommends the attachment of the gear with a bush rather than a keyway. Whilst a keyway is a compatible element it would not be considered complementary or preferred.

As previously discussed in section 6.1 there is a high level of dependency and recursion between the various stages of phases two and three of the proposed design process, figure 6.1. Because of this dependency, supportive methods and tools for each of the respective tasks within these phases; geometric and performance considerations, component matching, complementary evaluation and optimisation, must be considered collectively. This really can only be achieved within a single modelling environment or by interfacing a range of computer based design tools.

6.4 Compatibility analysis in an integrated modelling environment

In order to provide for the verification of compatible elements as defined above, the approach of many modelling tools is to inspect the input and output parameter fields of connected components (or their governing models). This inspection ensures that that there is an exhaustive and exact match of output parameters to input parameters for the coupled components. In this work, these parameters can be considered to represent the performance and physical attributes of a component, such as loading or relative dimensions and motions. The application of such a scheme to a mechanical system is problematic, because components may demand a range of parameters that can be fully met by all, or only part of the available output parameters from other compatible components. An example of this, is in the case of a shaft and bearing. Here the design data referring to the power and torque transmitted by the shaft is redundant, i.e. it is not required for execution of the bearing representation, in this case an electronic catalogue. Consequently, the parameters required by the bearing are fully met by only part of the possible parameters from the shaft. Hence, there is not an exact and exhaustive match of parameters between two compatible components in the mechanical domain. Therefore, compatibility cannot be determined through exact matching of parameters between two mechanical components, or indeed partial matching, which would be adequate in the case previously described. If the partial matching approach were adopted and a bearing was considered compatible with a shaft on the basis that all the bearing's input parameters can be met by only a proportion of the shaft's output parameters, then as a consequence, the bearing would be unable to distinguish between a shaft and a gear, which possess a similar range of parameters.

The fact that the compatibility of connected components in the mechanical domain cannot be determined by purely whether they have exact, similar or partial correlation between parameter is one of the challenges addressed by the work. In order to distinguish between compatible and incompatible components, permissible combinations must be explicitly defined within the modelling approach. This strategy is further frustrated by the inherent complexity of a mechanical systems representation, discussed in chapter 4 and shown in figure 4.10.

The requirements for compatibility analysis include the need for support during the configuration and construction of a design solution, and the explicit evaluation of the system for compatibility and matching once constructed. In order to provide for this latter requirement analysis routines must be executed during the system resolution process. To develop the requirements for this, it is first necessary to understand the strategy for resolution, which involves the propagation of data through the system model and the subsequent execution of component representations.

The modelling approach developed in this work adopts an elemental approach, where each elemental is a constituent of the system model. These elements represent mechanical components, parts such as mounts and system inputs/outputs. The data propagation cycle implemented has two distinct phases; firstly data is propagated inwards from the outlying elements, referred to as unitary elements, to the principal elements, following this the outward resolution phase propagates data from the principal elements to the unitary elements. Unitary elements possess only a single connection and are the outlying system elements such as inputs, outputs and ground points. Principal elements possess more than two connections and are the core components through which all other components or component chains are connected, such as a shaft. These elements demand full data propagation from all the connected elements in order that all the inputs are available prior to the interrogation and execution of representation. This two stage propagation cycle is necessary in order for component data fields to be fully populated and for the system model to be successfully resolved. However, the bi-directional nature of this cycle also complicates the incorporation of compatibility analysis in the approach. The order of resolution for a system is shown in figure 4.10, and a more detailed example is shown in figure 6.3. For this example the bearing is coupled to the shaft during the inward phase of resolution, whilst during the outward phase the shaft is coupled to the bearing. Permitting a bearing to shaft coupling or vice versa would satisfy both these conditions. However, such a condition would also be satisfied in the case of the component configuration shown in part (c) of figure 6.3, where the bearing is connected in between two shafts, which is incompatible. Consequently, the explicit definition or specification of permissible component combinations for both the inward and outward resolution phases is necessary. To provide for the specification of compatible components a 'knowledge base' is implemented within the modelling approach. This knowledge base can be interrogated during the model construction phase as well as being explicitly searched during resolution. The architecture for this knowledge base and its functionality are discussed in detail in the next section.

6.5 Compatibility knowledge base

The requirements for the compatibility analysis of mechanical components within an integrated modelling environment were introduced in the previous sections. In order to fulfil these requirements a knowledge base of some form is required. A knowledge base is defined as a formal and explicit representation of the knowledge pertaining to a given domain (McMahon & Browne, 1993). In this work, the knowledge needs to describe the compatibility of component combinations. The structure developed for the knowledge base is a multi-dimensional matrix, depicted in figure 6.4. This matrix provides for compatibility analysis during the bi-directional phases of system resolution, as well as providing for the distinction between compatible, incompatible and complementary component combinations. This distinction is achieved through the adoption of a weighted compatibility identifier which can assume a value of either 0, 1 or 2. An identifier is specified for each possible component combination. A value of '0' denotes an incompatible pair whilst a value of '1' indicates a compatible pair and '2' denotes a component pairing that is both compatible and complimentary.

6.5.1 A two-dimensional matrix for compatibility analysis

In order to provide for compatibility analysis during the two phases of resolution a two-dimensional compatibility matrix can be incorporated into the architecture of the modelling environment. The incorporation of the knowledge base enables the customisation of compatibility matrices by the designer to accommodate their individual preferences for manufacturers or suppliers. During each phase of resolution the matrix can be concurrently referred to in order to evaluate component combinations. To account for the change in order of data propagation and resolution, the search order for the matrix is transposed for each phase of resolution. For the inward phase of the resolution cycle the search order evaluates the i-j dimensions, this is inverted for the outward phase to the j-i dimensions, shown in figure 6.4. This example matrix contains seven component types and demonstrates the use of the complementary weighting between various types of generic components, such as bearings. In the case of a bearing on a shaft, bearing 2 is the preferred choice for reasons previously discussed, although bearing 1 is still a compatible alternative. The matrix system therefore provides for the inclusion of this preference, although the system does not capture the reasons (intent) for this precedence. Possible reasons for this may be associated with reliability, cost or favoured suppliers, and affords an area for future work.

6.5.2 A three-dimensional matrix for compatibility analysis

The two-dimensional matrix described above, addresses compatibility issues between component pairs, however, it does not suggest compatible structures, although it will evaluate structures by inspecting consecutive component pairings. The ability to suggest component structures is a desirable feature for any design support tool. Such a feature enables a designer who does not possess any knowledge of preferred components or suppliers to configure a system that comprises preferred or recommended components. By adding a virtual third dimension to the matrix, permissible chains or structures may be represented and displayed to the designer. This third dimension is achieved by searching consecutive instances of the compatibility matrix, illustrated in figure 6.5. If the designer selects component type 'A' from instance (i) of the matrix then evaluation of instance (i+1) for component type 'A' will generate a list of all possible compatible components. This approach can be extended to inspect instance (i+2) for each component derived from instance (i+1). In this manner, all possible component sequences can be displayed to the designer as a hierarchical tree structure, shown in figure 6.5 and for the case example in figure 6.8. As the designer proceeds through the design space committing to particular components so the number of feasible combinations reduces. This approach can be used to identify all compatible components or limited to complementary (preferred) components only. Thereby ensuring an achievable design solution that takes account of the preferences of the respective design team or organisation.

6.5.3 Dynamic manipulation of the compatibility knowledge base

The incorporation of a three-dimensional matrix provides for the evaluation of compatible and preferred components as well as enabling the identification and suggestion of possible component sequences. This approach is adequate for compatibility decisions undertaken at a component level and explicitly defined in the matrix, however, as previously discussed in section 6.3, decisions on compatibility and matching may depend on the inspection of individual component parameters. This parameter or attribute dependency means that certain components may only be compatible given acceptable coupling conditions, both in terms of types and magnitudes of coupling parameters. An example of such a coupling parameter is between a shaft and a roller bearing. In this case, all types of roller bearing from different manufacturers and suppliers, are compatible elements for a shaft. However, some may not support axial loads, or shaft deflections beyond a certain value. Consequently, components may only be considered to be well matched if corresponding parameters between them are within certain ranges or exactly matched. These exact matches may be numerical values as well as textual properties, such as lubrication type or material.

The provision for both the specification and the inclusion of compatibility conditions is achieved by manipulation of the compatibility matrix. The procedure for this manipulation is shown in figure 6.6. The designers' input to the process is to construct conditions or equalities between attributes for related components or between a predefined value and a particular attribute. These conditions are evaluated and the matrix elements are updated to reflect the change in condition. This evaluation and manipulation of the matrix occurs after the system has been resolved, at which stage all component models have been interrogated and complete specifications for components determined. This enables the evaluation of parameters for individual components or between component groups against the predefined conditions, set by the designer. In this manner, any component(s) which do not satisfy the matching conditions are identified and the designer is prompted to take the appropriate action.

6.5.4 An integrated modelling approach with compatibility analysis

The integrated modelling approach developed in this work enables the representation of engineering systems for their embodiment with standard components. The approach expedites the concept to embodiment phase of the design process, enabling a solution principle, termed a conceptual schema in this work, to be entered into the system. This schema describes the *class* of mechanical component; such as a gear, a shaft or a bearing, and their arrangement in terms of connectivity, similar to a concept sketch. For the example considered, the conceptual sketch and the associated schema are depicted in figure 6.7 parts (a) and (b) respectively. In addition to describing the class of mechanical component, each icon also provides a container for a computer based model or electronic representation. This representation governs the design and selection of a particular component type and is assigned by the designer. The electronic representations range from electronic catalogues, numerical codes, spreadsheets and databases to models constructed in CAD systems. It is at this stage that the designer commits to a particular supplier or *component type*. Where this *component type* represents a specific range of mechanical elements from a particular manufacturer. It is during the construction of the schema that the first level of compatibility analysis can be invoked. This supports the selection of component types, sequences and importantly component manufacturers or suppliers. To achieve this, the three-dimensional matrix discussed in 6.5 is utilised. This provides a mapping of the possible components sequences, which can be coupled to a particular component and suggests compatible components and/or preferred components. For the example considered in figure 6.7, the compatibility matrix is invoked during model construction for the shaft and the keyway. This process automatically references the compatibility matrix and produces a dendritic structure, which describes the various component sequences that are capable of being coupled to each component. For the case of the shaft and the keyway, the possible component arrangements are shown in figure 6.8 part (a) and part (b) respectively.

The second level of compatibility analysis occurs after the system has been resolved and involves the dynamic manipulation of the matrix. During this process, certain attributes between related components are evaluated. If conditions are not satisfied then the component pairing is deemed incompatible and the designer is notified. An example of this might be the shaft deflection and the maximum allowable angle of misalignment for a bearing. The angle of misalignment for a given bearing tends to be range specific. That is, the value of the attribute is constant across the given range. Consequently, static attributes such as these are evaluated after the system resolution phase, at which stage values for the various attributes of a component are available for comparative assessment.

These conditions will have been specified in advance by the designer by virtue of the 'constraint specifier', depicted in figure 6.9. Constraints between components are specified as logical conditions (<, >, =), which are evaluated post resolution. These conditions are generally expressed as numeric equalities, although string comparisons can be evaluated, but this is restricted to exact matching only. This feature can be used to ensure that attributes such as lubrication types or materials are consistent throughout a design. If the constraints are not satisfied the designer is notified and prompted to take the appropriate action, in order to resolve the conflict and satisfy the constraint.

6.6 Concluding remarks

This chapter discusses the important issue of compatibility analysis for standard components during system modelling. Three aspects of compatibility analysis are developed: *connectivity analysis, performance matching* and *complementary assessment.* A review of compatibility analysis techniques in other engineering domains is also undertaken and the associated issues for the generation of compatibility analysis in mechanical systems are discussed.

A strategy for compatibility analysis within an integrated modelling approach is developed that provides for the distinction of compatible and incompatible components as well as for the utilisation of preferred combinations of components, such as shafts and particular bearings or gears and keyways. The incorporation of compatibility analysis ensures that 'achievable' designs are configured and through complementary evaluation makes certain that 'preferred' components are utilised where possible. Achievable designs are those which are configured from components that can be procured exactly as specified, usually these will be standard components, and are matched such that all components are physically connectible and mutually complement each other in terms of the type(s) and magnitudes of their inputs and outputs. This ensures that the design can be produced and that it will operate within the performance capabilities for which it was designed. The ability to configure achievable systems within a modelling environment is invaluable for the early stages of system design and for rapid development. Furthermore, for complex systems comprising tens or even hundreds of components, this type of assessment is essential.

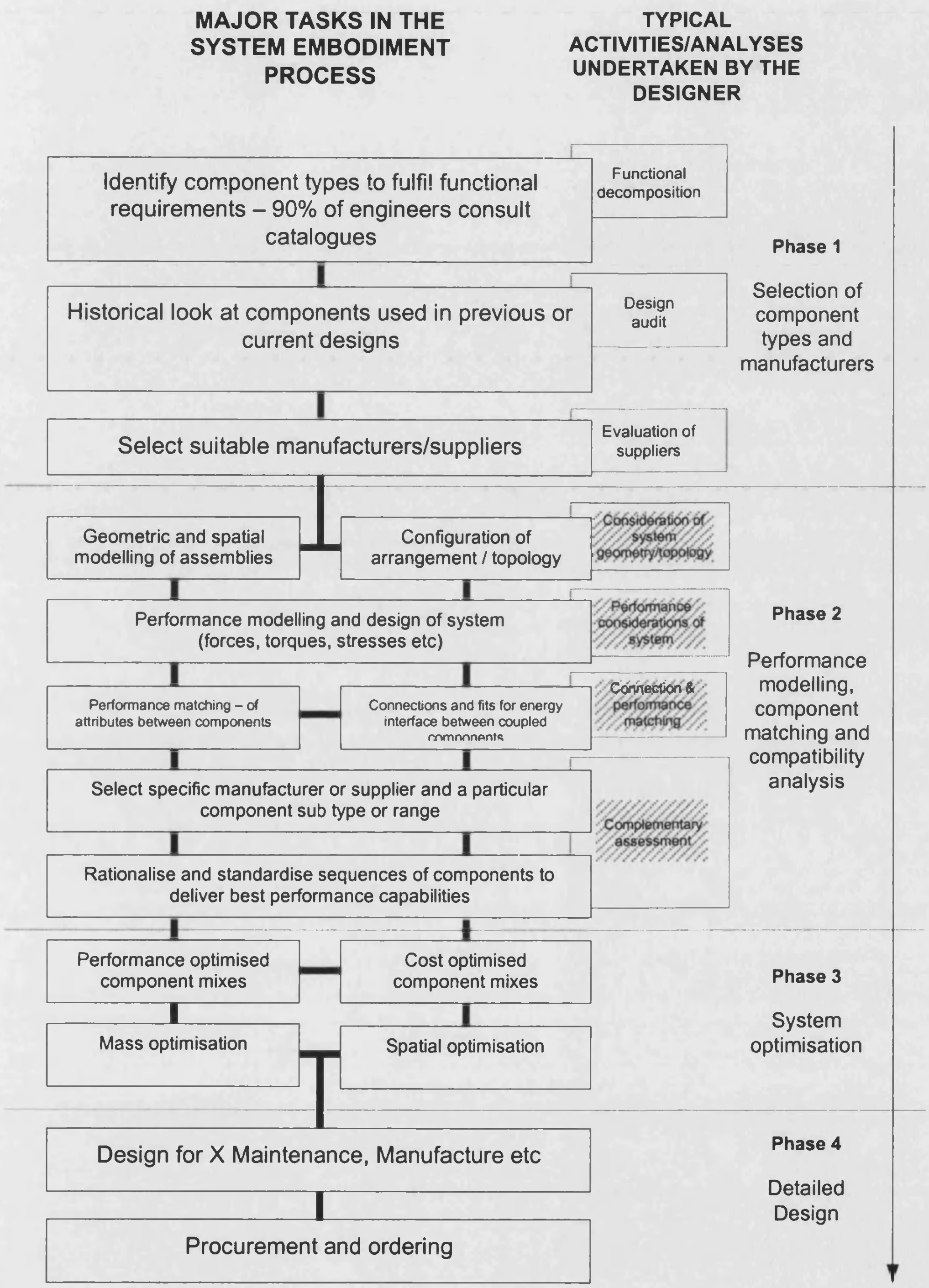


Figure 6.1 – Design process for the incorporation of standard components

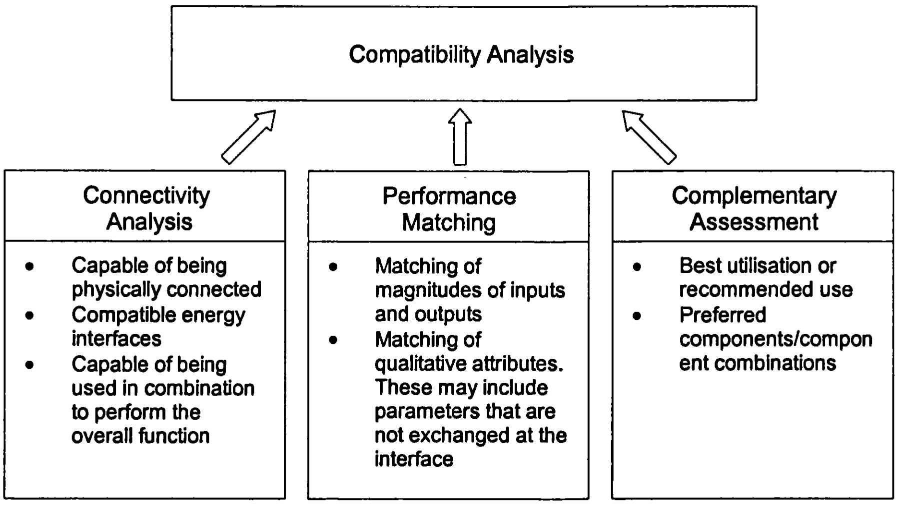


Figure 6.2 – The proposed functions of compatibility analysis in mechanical systems

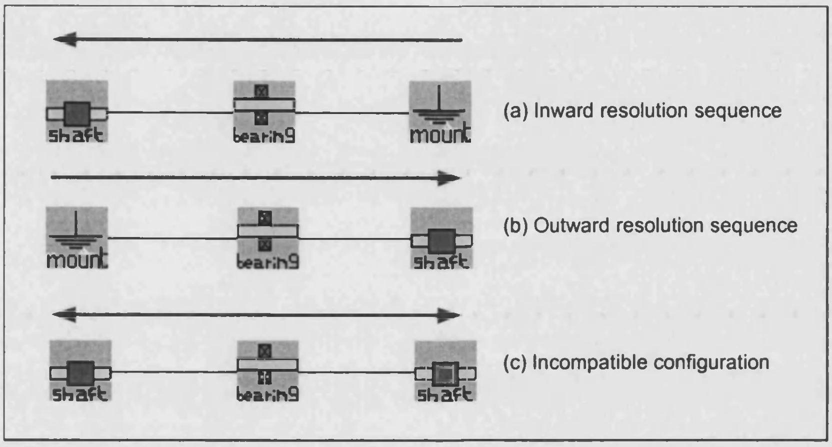


Figure 6.3 – Order of resolution through a possible component sequence

i - dimension

J - dimension

	Gear1	Shaft	Bearing1	Key	Bush	Bearing2	Gear2	...
Gear1	0	1	0	0	1	0	0	..
Shaft	0	0	1	1	0	2	0	..
Bearing1	0	0	0	0	0	0	0	..
Key	1	0	0	0	0	0	1	..
Bush	2	0	0	0	0	0	1	..
Bearing2	0	0	0	0	0	0	0	..
Gear2	0	0	0	0	0	0	0	..
...	..	..	..	..	..	..	..	..

Inward search order (i-j)

Outward search order (j-i)

Figure 6.4 – A two-dimensional compatibility matrix

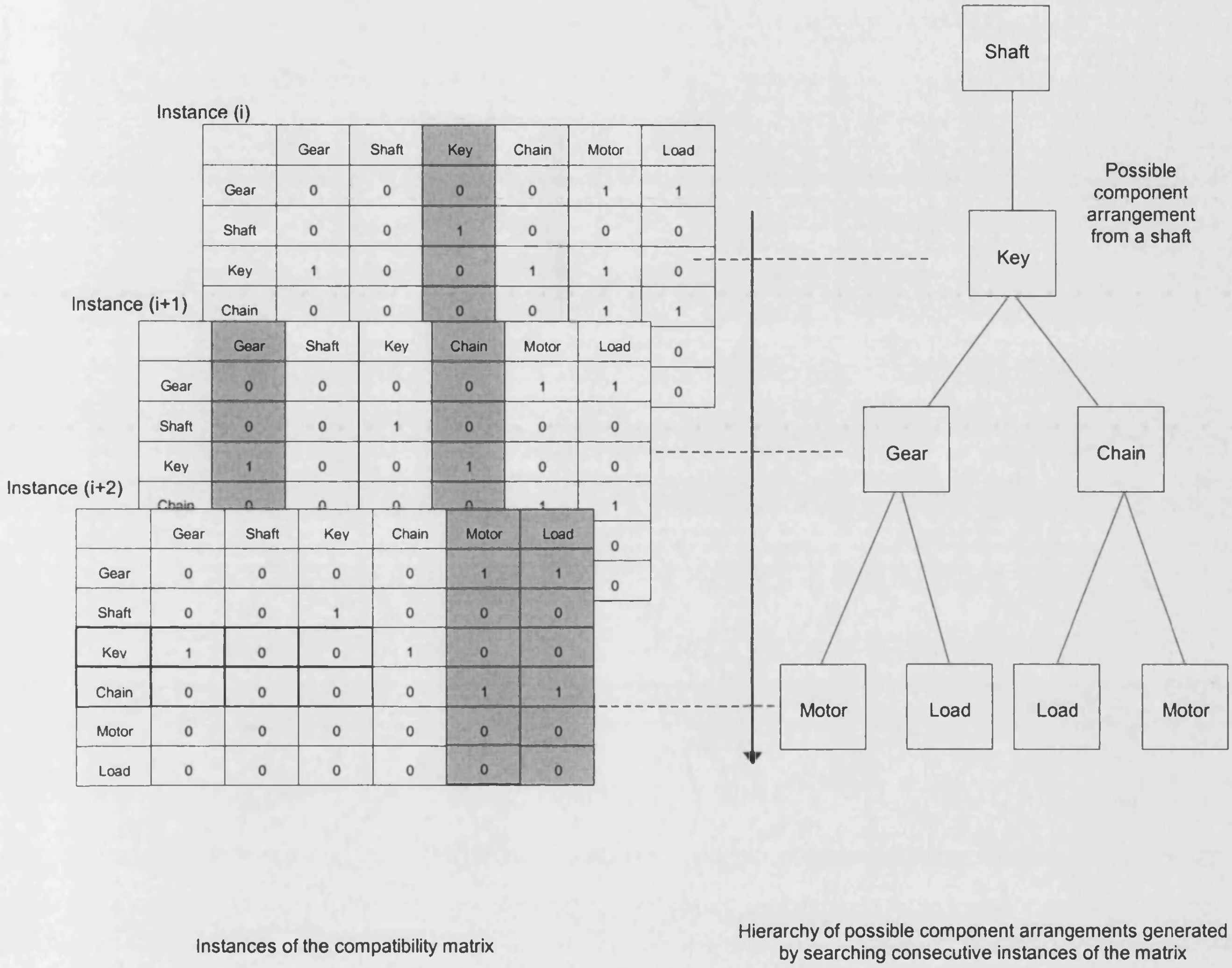


Figure 6.5 – A virtual three-dimensional compatibility matrix

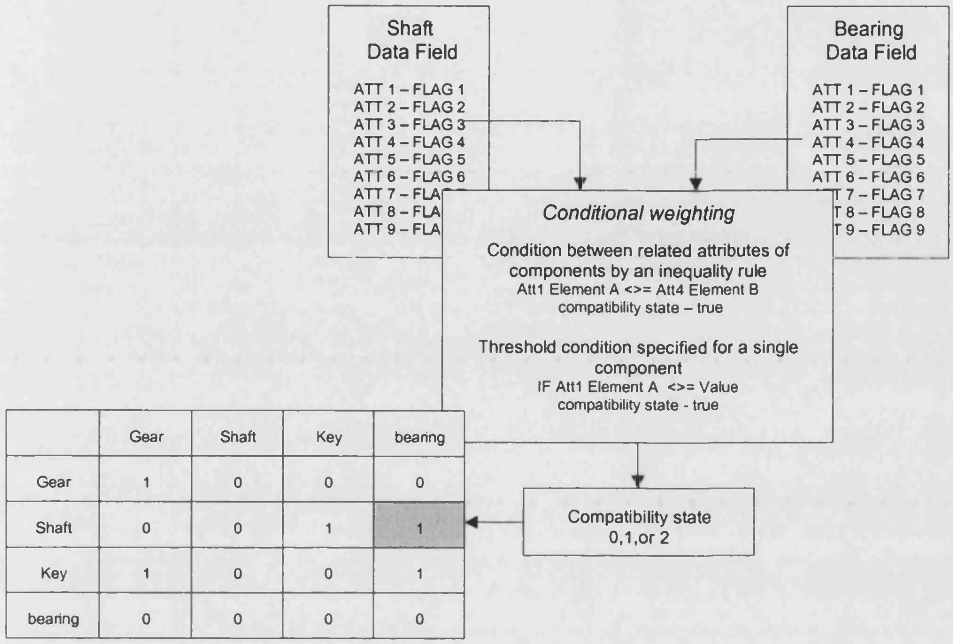
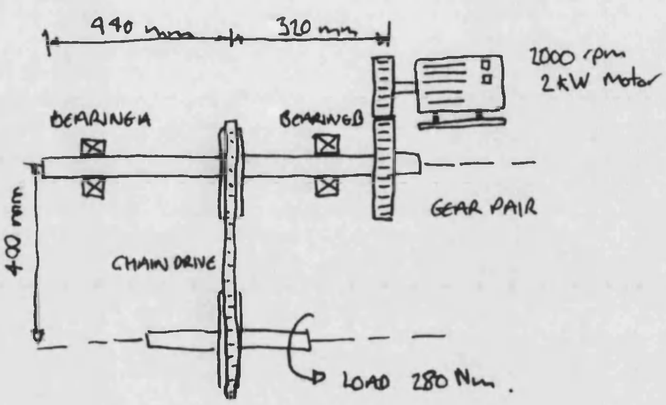
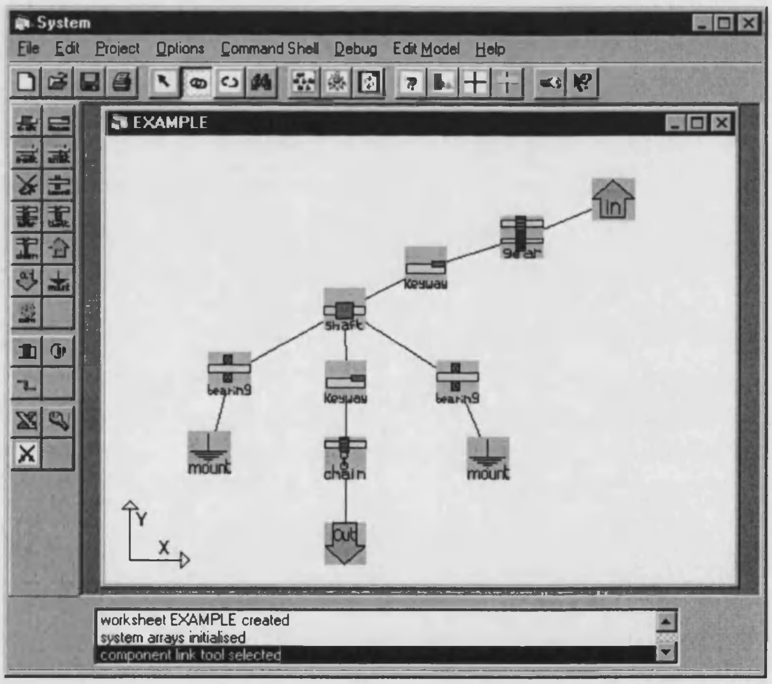


Figure 6.6 – Dynamic matrix manipulation by inspection of component parameters

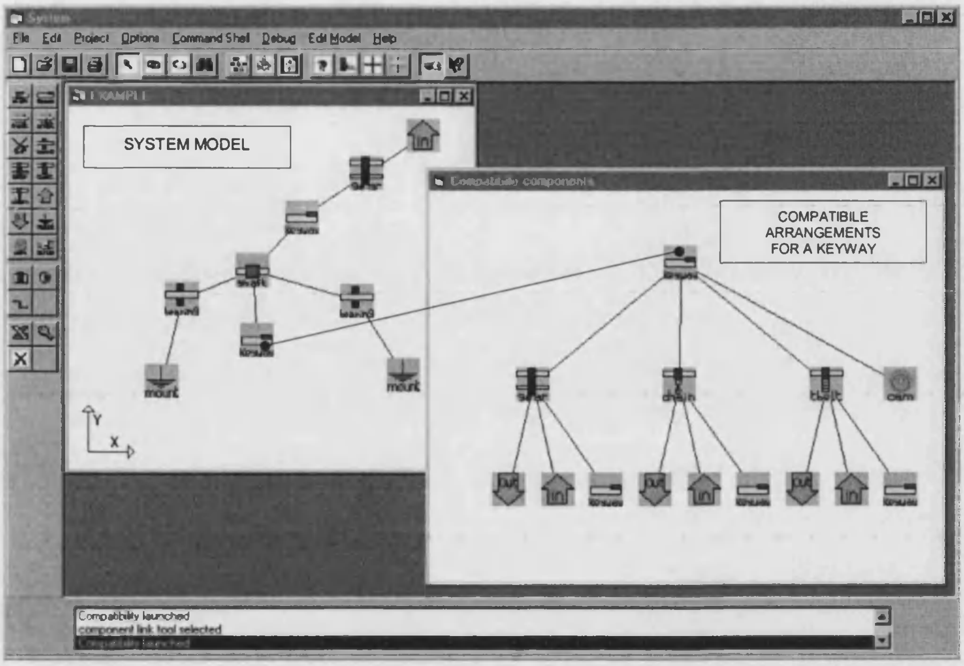


Part (a) Concept sketch

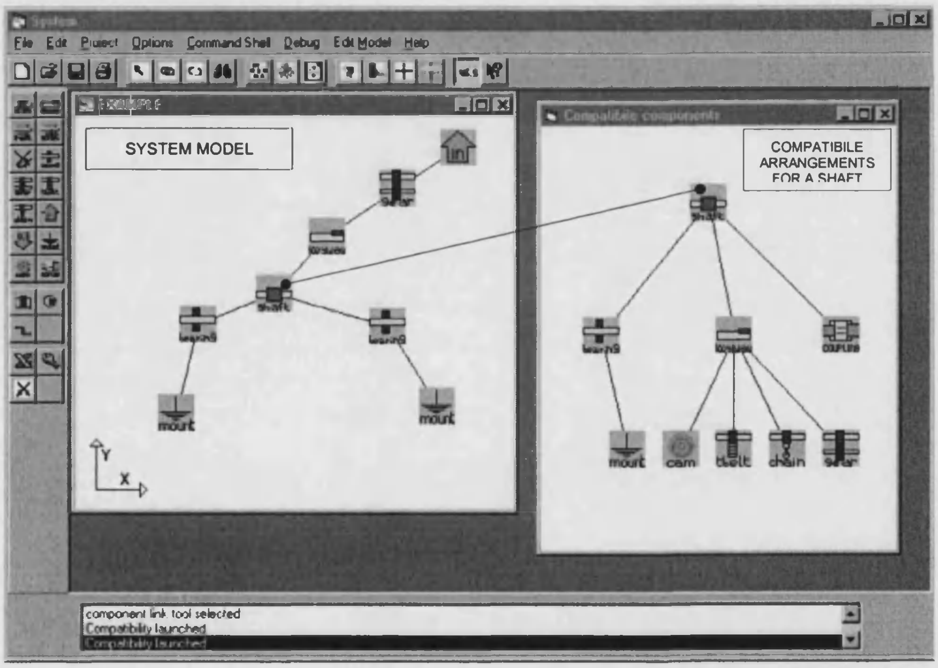


Part (b) Concept schematic

Figure 6.7 – A connectivity model of a concept principle in an integrated modelling environment



Part (a) Compatible arrangements for a keyway



Part (b) Compatible arrangements for a shaft

Figure 6.8 – Compatibility analysis during model construction

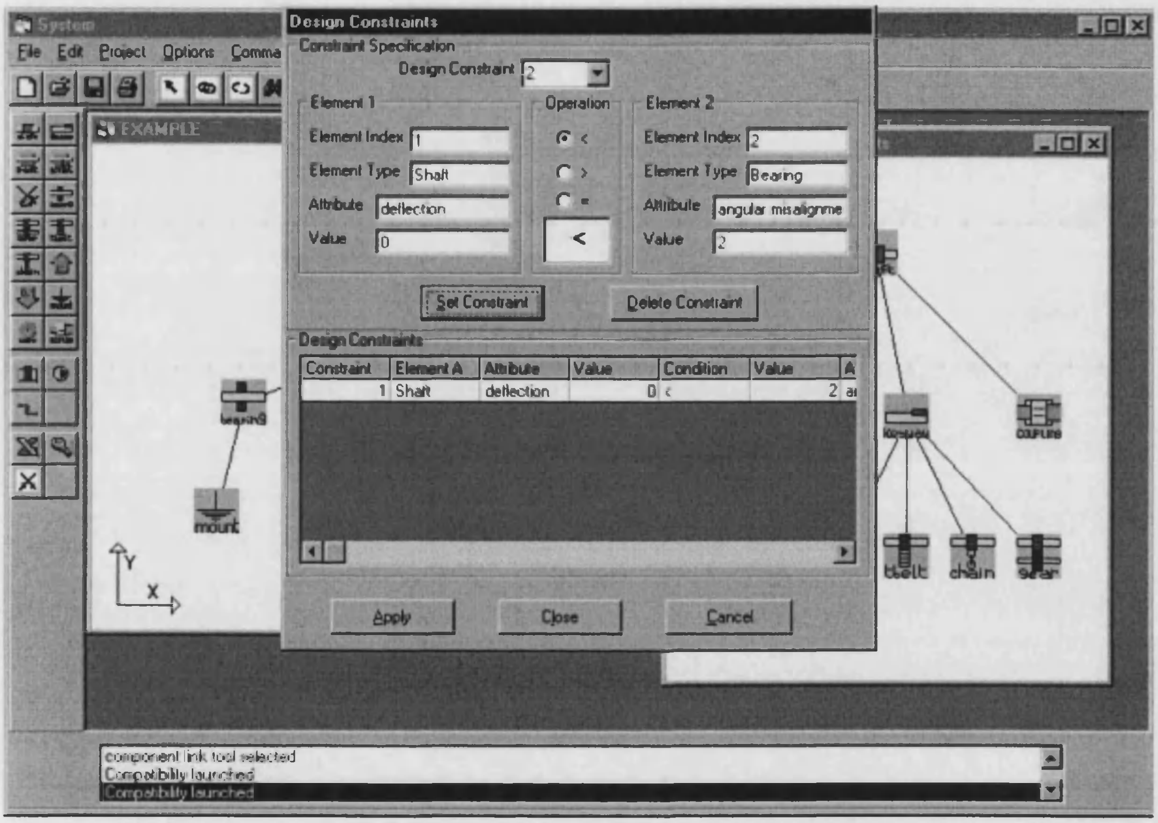


Figure 6.9 – Specification of compatibility constraints between related attributes

Chapter 7

Interfacing electronic representations for system modelling

The importance of standard components in engineering has been discussed in chapters 1 and 2. Furthermore, the benefits of electronic representations for the selection and design of engineering components have also been highlighted. A review of the various types of electronic representation was undertaken in section 3.2 and the various classes of electronic representation for standard components are shown in figure 7.1. These electronic representations provide very powerful tools for the design, selection and specification of individual engineering components. This is because the various technologies incorporated into the representations have been developed over many years to deliver improved representation of the components and better match the needs of the designer. These needs are documented by authors such as Webber (1994) and include: advanced search techniques, supplementary information, costing and availability. For the purpose of this work, the electronic representations dealt with are component based representations. These component based representations describe a particular type of engineering component and enable the design, selection and specification of a particular component configuration from the range available. Such representations are available for many different components and include, for example, the design of shafts, gears, the selection of bearings and the selection of motors.

The high level of dependency, iteration and recursion involved in a systems approach for embodiment design is depicted in figure 7.2. For each iteration the appropriate representation governing each component must be activated and interrogated by the designer. In fact, current design practices require the designer to undertake many of the time-consuming data manipulation and searching operations manually. The considerable level of resources required, frustrates the ability of the designer to determine an optimum solution. It is therefore highly desirable to develop methods which free the designers' time and in particular develop a systems approach that integrates the various classes of electronic representation for their automatic operation. The realisation of such an approach enables the embodiment of complex mechanical systems from reliable and accurate third party component representations. It would also remove many of the repetitive tasks of data entry and software execution, enabling the designer to spend more time on developing and refining design solutions.

The incorporation of third party component based representations in a systems approach ensures that real components are considered. These 'real' components enable the actual values for mass, cost and spatial occupancy to be generated and more fully informed decisions to be taken by the designer. The term 'real' denotes elements which may already exist, have been previously used and have predetermined or predefined properties. In many cases, these are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified. The inclusion of real components in a systems modelling approach is essential, especially where optimisation techniques are to be applied. This is because of the high level of dependency between components in a mechanical system. This dependency relates to physical dimensions and performance capabilities, which must all be manipulated in order that a system of components is matched in terms of connectivity and performance. Furthermore, any alterations in the specification of an individual component may well demand changes in connected components in order to accommodate the changed part, and such changes may well produce a significant change in the performance of the optimised system, especially for sensitive solutions. It is therefore important that real components and their associated 'real data' is used. In this manner, an optimised set of components can be procured exactly as specified by the system.

7.1 An integrated modelling approach

The development of a mechanical systems modelling approach, discussed in chapters 4, 5 and 6, enables the coupled nature and dependency of connected mechanical components to be represented for system embodiment. The automation of this embodiment process enables the designer to consider many more configurations, component types and sizes within a shorter time. This enables the designer to develop a better performing solution, which might not have been obtainable if the more traditional, manual procedures for embodiment were used. If only because of the considerable timesaving. The system modelling infrastructure developed in this work enables the construction of a model that represents the connectivity and dependency of components within a system, illustrated in figure 7.3. This approach also separates the system representation from the component representations. This is achieved by implementing two sets of data for each element within the model: a complete set of performance attributes and design constraints (imposed by connected components) and a parametric data set. This parametric data set is generated from the performance attributes and design constraints and affords the selection data for the interrogation of the appropriate component representation, an overview of this strategy is shown in figure 7.4.

In order to consider 'real' components the modelling approach has to accommodate the wide variety of representations for the standard classes of mechanical component. These include databases, spreadsheets, numerical codes and CAD based representations. Further to this, the diversity of modelling approaches used to represent both continuous and discrete component ranges as well as varying levels of modelling abstraction make the development of methods that provide for the incorporation of these representations within a modelling approach problematic.

7.2 Incorporating third party electronic representations

For the purpose of incorporating third party electronic representations into the modelling approach two strategies are considered; the first deals with the generation of standardised models from existing representations, whilst the second deals with a standardised approach for interfacing these electronic representations with the modelling infrastructure.

The approach of generating standardised models from existing representations was adopted for assembly modelling by Theobald (1995) and for developing a large homogenous database of catalogue components by Vogwell & Culley (1991). In the creation of a large homogenous database for bearing selection all performance data was entered manually in order to populate the database. This is both a time-consuming and inefficient method especially where component types, ranges and availability change rapidly. For the modelling of engineering components, Theobald (1995) focussed on the development of parametric models for particular component types and ranges. However, the development of the models proved difficult for a number of reasons:

- For certain components the development of mathematical relationships between performance parameters across the included range is difficult. This is because for many selected components, performance parameters are determined empirically, and hence rarely follow a straightforward mathematical relationship.
- Many manufacturers and suppliers rationalise their available range, driven by the demands of the customer. This often leads to a non-uniform discretisation of components across the range, which can only be approximated by continuous parametric models (Theobald, 1995).

Through evaluation of this approach a number of serious limitations are identified. These relate to the usefulness of the approach in determining a physically realisable or optimum system of mechanical components.

- Abstracted models must be altered or updated every time a manufacturer or supplier change their included range of components.
- The often non-uniform discretisation of components cannot be reliably described, unless abstracted database techniques are implemented.
- A model must be created for each component type and the various manufacturers. This model construction process requires knowledge of advanced modelling approaches such as polynomial regression and neural networks.
- The additional functionality of the original component representation must be included in the model, otherwise it is lost. This additional functionality (added value) arises because individual component representations are focused on the component and respective supplier, and have been tailored by developers to meet the individual requirements of the designer (user) and component manufacturer.
- Abstracted models cannot contain up-to-date accurate information on availability and costs, which are commonplace in many web based and hybrid representations (Allen *et al,*2000).

In addition to these disadvantages, the creation of abstracted models seriously frustrates the development of an optimal system of mechanical components. If approximate performance data is used and the optimised solution is sensitive, then when actual components which are the closest match to the approximations are procured, a sub-optimal design is more than likely and in extreme cases a poor performing solution may be developed.

Many of these disadvantages are brought about through not utilising the actual component representation developed by the supplier or manufacturer and are overcome in the second approach. This approach involves the development of a standardised procedure for interfacing electronic representations. However, whilst this approach provides many benefits for the designer it demands the resolution of a number of issues for its successful implementation. These issues involve the development of a protocol for transferring essential data between component representations, and procedures for the remote operation of component representations. These issues are dealt with in more detail in the next section.

7.3 Interfacing electronic representations

In order to interface component representations, this work has identified four key features necessary for interaction between the modelling environment and the electronic representations. These are:

1 The ability to control or activate the electronic representation.

2 The ability to interrogate and invoke commands within the electronic representation.

3 The ability to exchange information between the modelling environment and the representation.

4 The ability to construct information and respond meaningfully to information exchanged.

The first three of these requirements are dependent on the capabilities of the software, at both an operating system level and an application level. Whilst the latter requirement depends on the electronic representation, and in particular the respective codes of practice used for software development, software structure, interface design and conformance to standards for representing data describing engineering components. Although as is discussed in a later section there are few standards for describing information that represents engineering components, other than for purely geometric purposes such as STEP (PDES Inc, 2000).

7.3.1 Architecture for interfacing electronic representations with the system model

An investigation of the capabilities of current technologies incorporated in computer operating systems reveals that current software capabilities can provide for requirements 1 to 3 detailed earlier. Within the Windows operating system two protocols are supported that allow data and commands to be exchanged between applications. These technologies are Object Linking and Embedding (OLE) and Dynamic Data Exchange (DDE), both of which allow interprocess communication between software applications at runtime (Cornell, 1999). These features are commonplace in office applications such as Microsoft Excel and Lotus 123 and are often integrated into commercial CAD systems, such as SolidWorks or ProEngineer. However, many electronic catalogues and bespoke commercial codings are developed by one of a large number of small web/electronic publishing companies, resulting in software applications that do not conform to accepted standards or codes of practice (Microsoft, 2001a), in which cases remote operation is all but impossible, although automatic activation is still possible, as these events are invoked at an operating system level.

The Windows DDE protocol is a standard for cooperating applications. By using the protocol, applications can execute remote commands by means of Windows messages. One of the primary features of the DDE protocol is to perform data queries between applications such as a proprietary software module and a spreadsheet or a database (Microsoft, 2001a). In order to utilise the DDE convention, applications must contain the necessary functionality to participate in a conversation. This functionality essentially provides for the identification of the units of data to be communicated. This follows a three level hierarchy: Application, Topic and Item (Microsoft, 2001b). In this manner, text strings and values contained within objects such as the elements of a database or text box can be transferred. Furthermore, this technology enables commands (menu items) to be invoked remotely. This enables data to be exchanged, applications to be activated and commands to be invoked which provides the basis for full remote operation.

Through the utilisation of this convention, standardised procedures for interfacing electronic representations are possible. However, the fourth requirement, which pertains to the ability to construct information and respond meaningfully to information exchanged, demands that either a common information representation is developed or that information is translated and parsed between applications, similar to the exchange of CAD information.

7.3.2 Information representation for interfacing electronic representations

The ability to meaningfully exchange information between the modelling environment and electronic representations is frustrated by the fact that the formats used by many small web/electronic publishing companies are proprietary and are all too often supplier focussed (Allen *et al*, 2000). Furthermore, there is a lack of standards or documentation for describing engineering component attributes and values. Such attributes include both the physical and the performance data necessary to fully specify and represent an engineering component. Examples include the internal diameter and load rating of a bearing, the lubrication type or pitch for a chain drive and the deflection of a shaft. These issues make meaningful exchange of information between various software environments by virtue of a single, common protocol, all but impossible. The development of a standard for representing component attributes for the communication between computer based representations is beyond the scope of this work and affords an important area for future work. However, to overcome these problems and to demonstrate the overall approach; software objects that translate and exchange information between the modeller and the various representations are implemented. The exchange and translation processes for these software objects are depicted in figure 7.5. In order to interface the different classes of electronic representation three generic *intermediaries* have been developed.

- An ASCII data file which comprises the component attributes and design constraints. This can be used as an intermediary for the exchange of data between applications that do not incorporate full DDE functionality.

- An ActiveX component that provides a DDE interface with an ACCESS database. This component provides for the generation of a parametric data set and the construction of a query for the interrogation of a database. This component also provides for the parsing of the results of the query to the modelling environment.
- A DDE interface with a standardised spreadsheet. This standard spreadsheet comprises the component attributes, design constraints and arbitration routines. This can be used to incorporate models that are based on a spreadsheet design environment as well as providing an intermediary with other applications such as CAD systems. The Excel environment provides additional functionality for controlling and linking to other Windows based applications. In this work, an intermediary spreadsheet is used to provide automated interrogation of a bespoke model constructed in SolidWorks.

These intermediaries provide for the activation, interrogation, exchange of data and the deactivation of the electronic representation. A schematic overview of these intermediaries is shown in figure 7.6. These intermediaries are generic and need to be customised for each particular component and associated representation. This customisation is necessary for the different number and types of component attributes and the assimilation of selection data, such as the search query for a database or the driving parameters for component selection algorithms. Furthermore, these intermediaries provide for the initialisation data for each component during the construction of the system model. An example implementation of a spreadsheet sheet intermediary for design of a gear pair is depicted in figure 7.7, with annotated details of its *modus operandi.* In order to demonstrate the overall modelling approach, intermediaries have been implemented for a bearing catalogue, chain drive selection software, gear selection spreadsheet, a shaft design model, a keyway design module and a model of a cam and follower constructed in SolidWorks. These electronic representations have been created to demonstrate the approach. An overview of the various representations is included in the Appendix. The software objects that provide the translation and compilation are incorporated into the software modules that provide the agent based arbitration, discussed in chapter 5. In this manner, a single software component provides for all aspects of model interfacing, including control of the representation, data exchange and data arbitration.

The approach adopted, although flexible does require designer intervention in order to tailor the generic intermediaries for new representations, which requires a detailed understanding of the electronic representations. And particular component attributes, selection data, search techniques, selection algorithms and how to invoke operations. This understanding is necessary because of the various conventions adopted by different manufacturers and suppliers.

7.4 Concluding remarks

This chapter discusses the various forms of electronic representation for mechanical components and describes an approach to enable these to be incorporated within a systems modelling approach. The requirements for interfacing electronic representations with a system representation are developed and a software architecture created for the inclusion of the various classes of electronic representation. Software intermediaries for each class of representation enable the modelling environment and the electronic representations to be linked dynamically at runtime. These intermediaries provide a protocol for the transfer of data between representations and the modeller, control of selection data and the remote operation of the component representation. This removes repetitive data manipulation tasks and provides more time for the designer to refine concepts. The approach and associated technology enable the full range of electronic component representations to be incorporated into a systems approach. Furthermore, the inclusion of these third party representations means that 'real' components are used throughout the process. Such an approach also provides a platform for optimisation procedures with a solution space that bounds real data provided by manufacturers or suppliers.

The work described in this chapter has confirmed a very important research issue to be addressed by future work. This involves the development of standards for the exchange of data describing standard engineering components. This would involve two distinct aspects; the development of a standard for data representing the performance attributes of standard mechanical components and a software architecture or procedure for communication or open access between electronic representations. The development of techniques which deal with these areas are essential for the searching of multiple electronic representations, perhaps from different suppliers, and for interfacing component based representations with systems modelling environments.

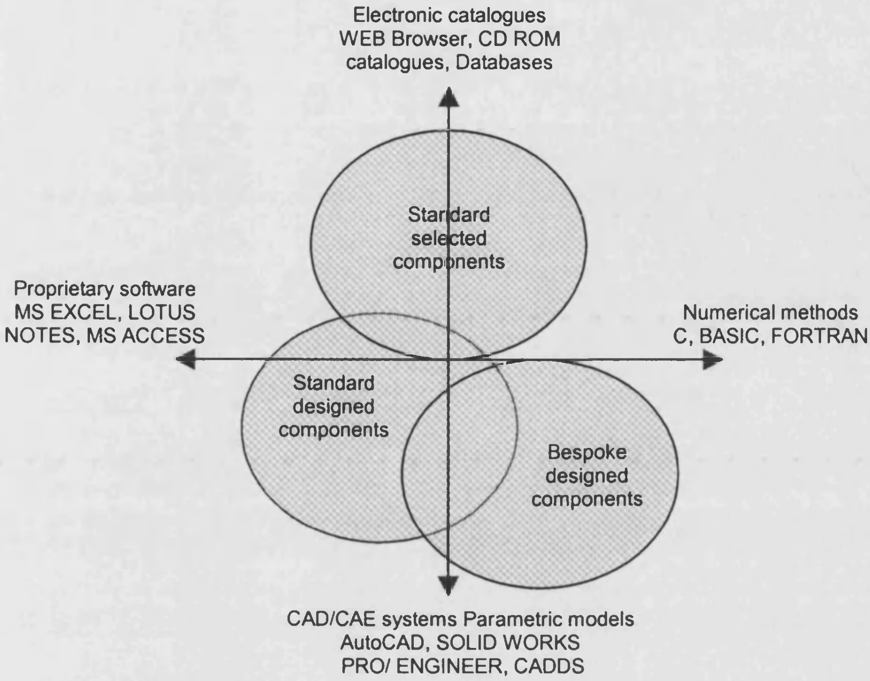


Figure 7.1 – Classes of engineering component and their associated form of electronic representation

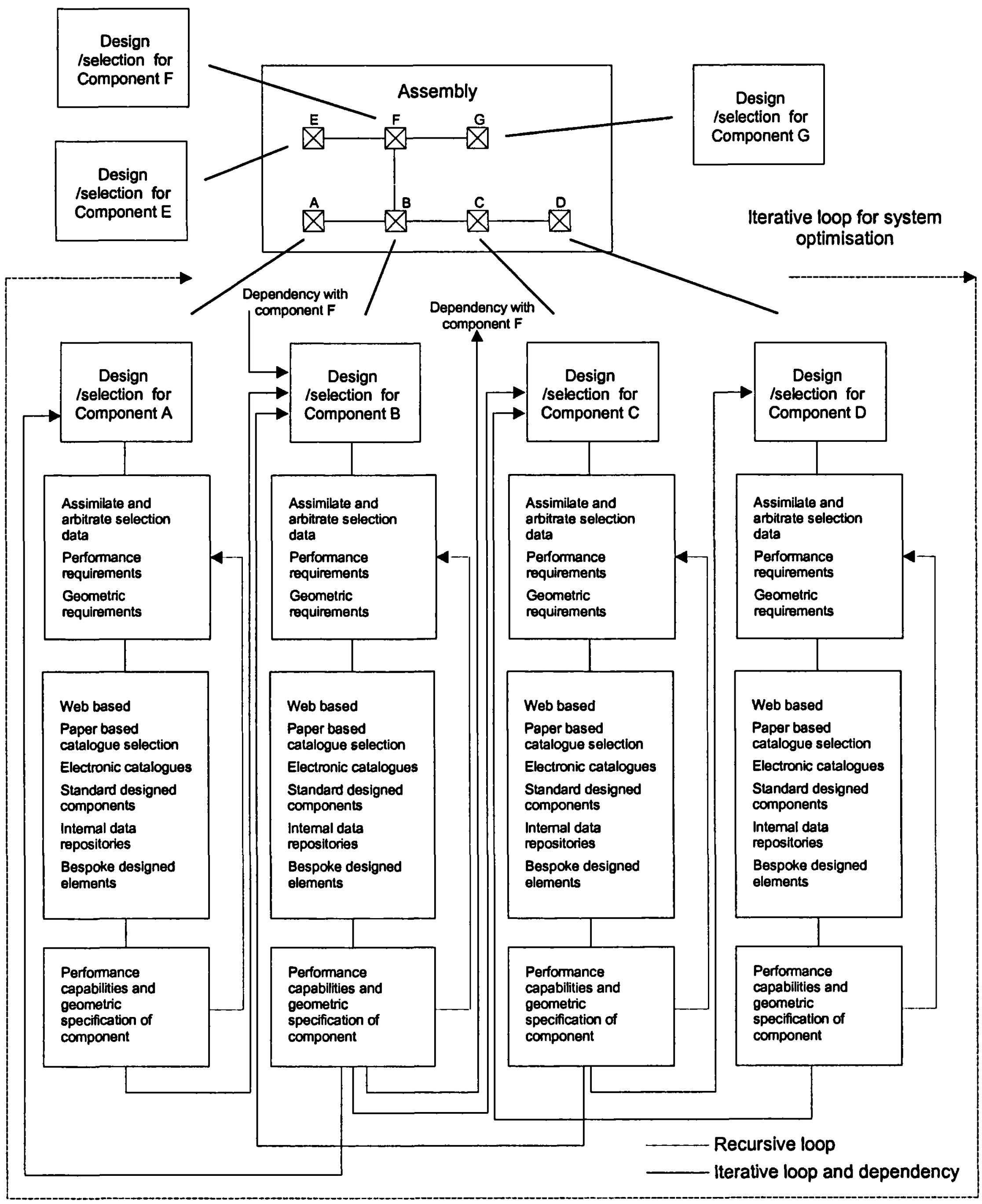


Figure 7.2 – A systems approach to component selection for part of an arbitrary assembly

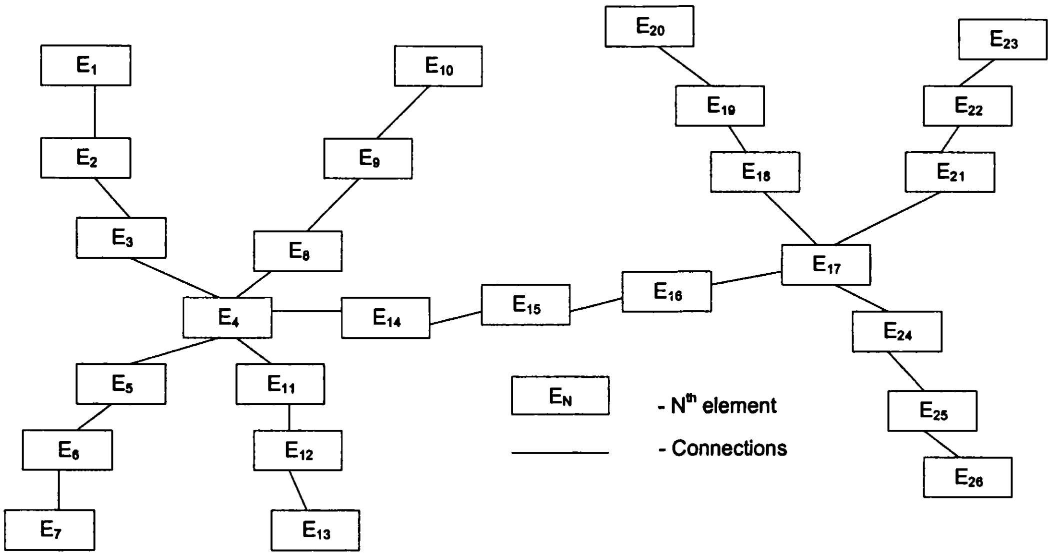


Figure 7.3 – Connectivity representation of a mechanical system

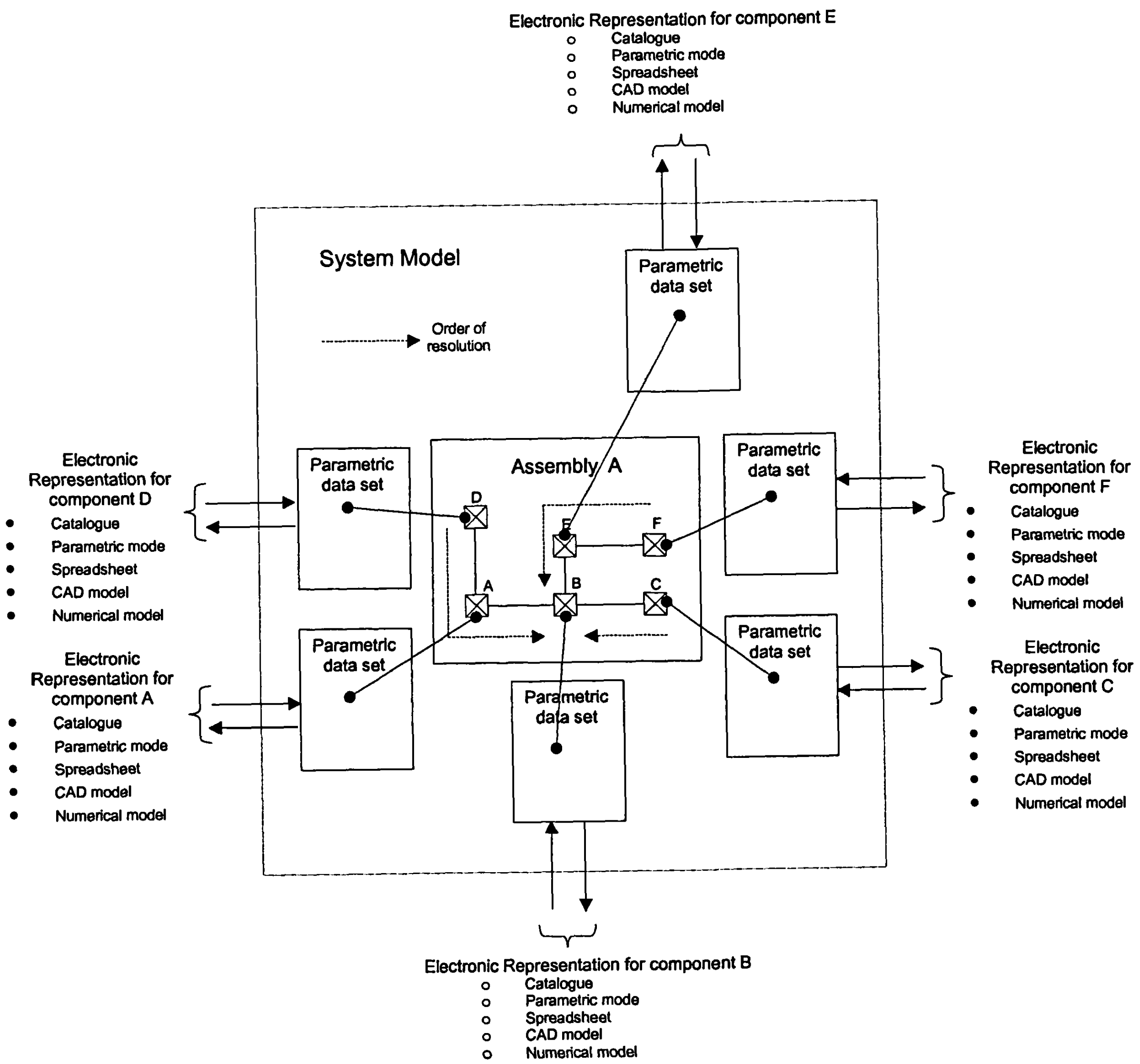


Figure 7.4 – Connectivity model for a mechanical system

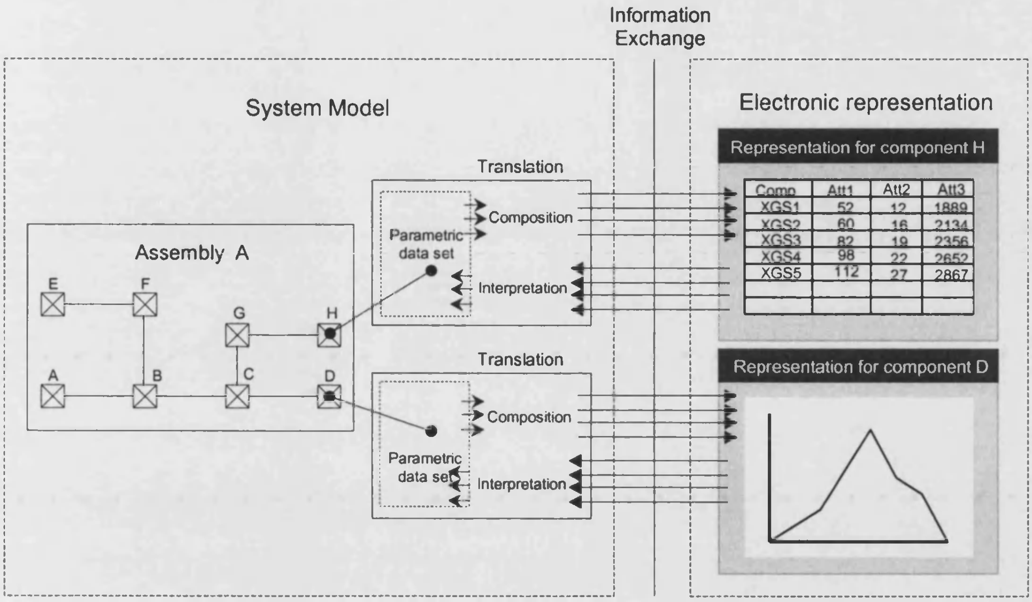
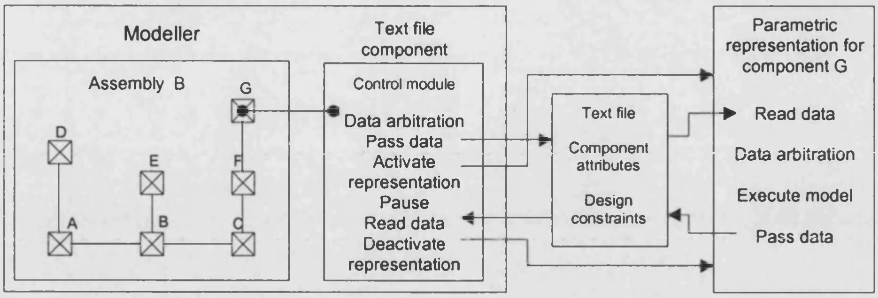
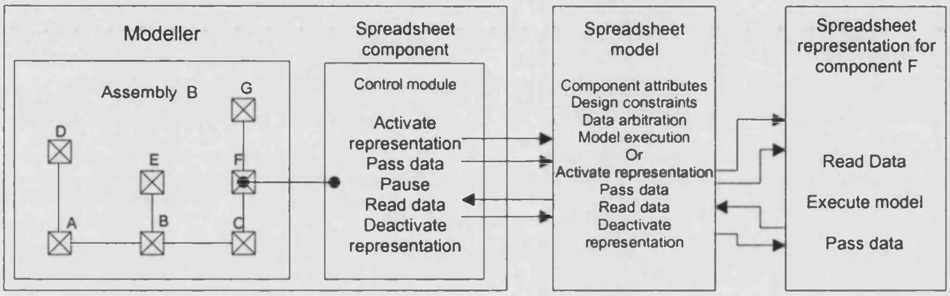


Figure 7.5 – Data exchange and translation requirements



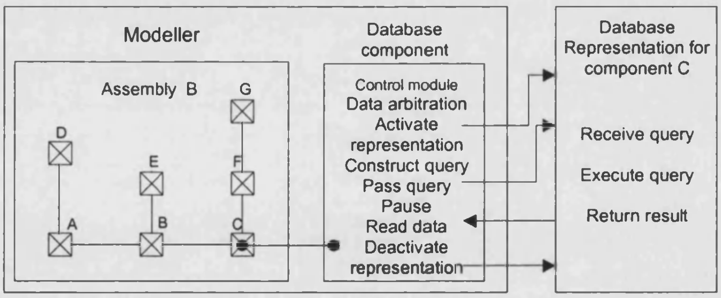
Note: Data arbitration can occur either in the modeller or within the component representation

Part (a) – Utilisation of an AS CII text file as an intermediary



Note: The spreadsheet can either contain a component representation or act as the control for another representation such as a CAD system. This enables the additional functionality of the environment to be utilised.

Part (b) – Utilisation of a spreadsheet as an intermediary and a control environment



Part (c) – DDE exchange for the interrogation of component databases

Figure 7.6 – Three types of intermediary used for interfacing electronic representations

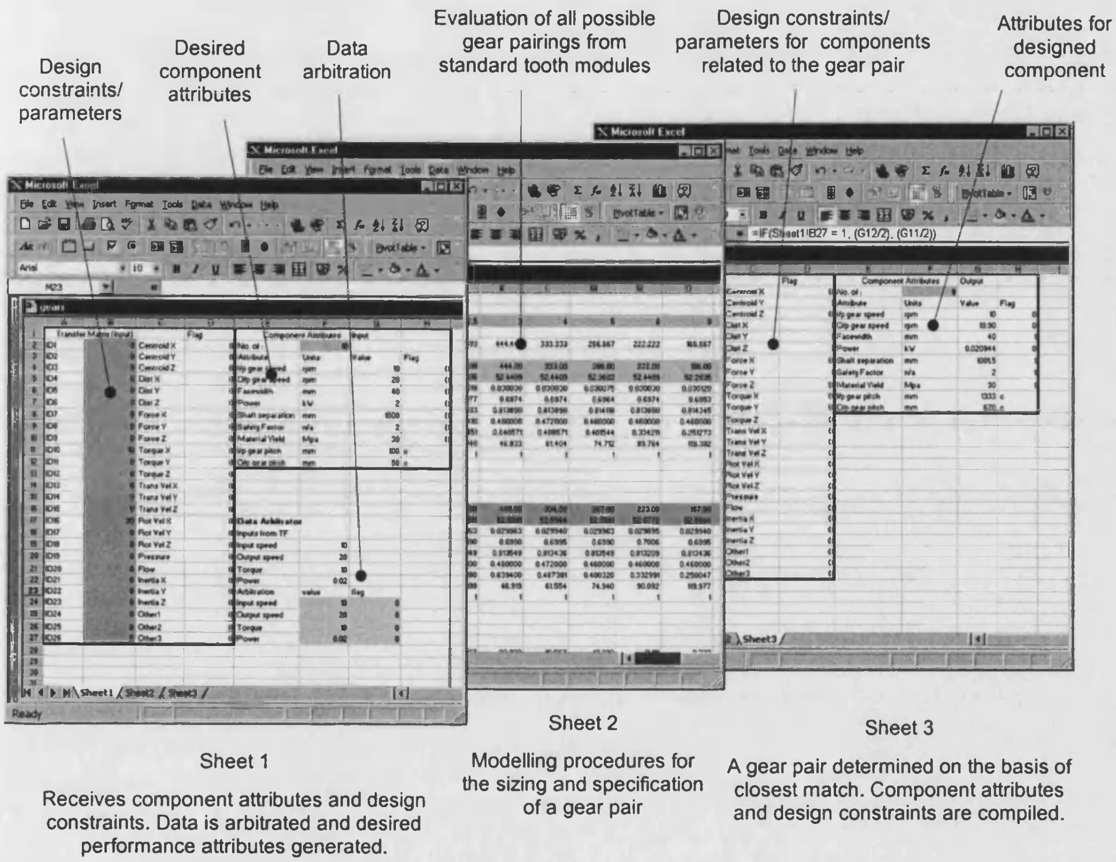


Figure 7.7 – A spreadsheet model for the selection of a gear pair

Chapter 8

An integrated modelling environment for mechanical systems

The configuration and embodiment of a design whether mechanical, electrical, hydraulic or pneumatic demands a significant amount of time and effort, which is distributed over a number of essential tasks. These include the specification and selection of individual components to deliver the desired capabilities for the system, as well as ensuring that individual components meet the performance and physical constraints imposed by their connected components. By their nature, these tasks are analytically intensive and demand many iterations before an acceptable final design is achieved. As a consequence, this process is ideally suited for automation by computer, especially where systems comprising standard components are considered. This is because standard components are typically only available over a limited range of sizes and therefore may require many iterations before an acceptable solution is determined.

During the embodiment of mechanical systems and in particular machine systems, the consideration of both performance and geometry is necessary in order to ensure that components are physically connectible and satisfy essential performance requirements, such as speeds, loading and torques. Indeed, computational methods for component identification and selection through performance modelling has been the subject of much research (Ashby, 1999; Culley & Webber, 1992; Richards *et al*, 1999). Attempts to integrate component models, driven by performance requirements, within a system modelling environment have been undertaken in systems such as: Schemebuilder (Counsell *et al*, 1999), AMEsim (AMESim, 2000), Spice (Keown, 1994) and Design Compiler (Ward & Seering, 1989). Modelling environments in both the fluid power domain and the electrical domain have been successful, providing excellent design through simulation tools. In contrast, attempts in the mechanical domain have been hampered by a number of problems. These include extensive hard coding of bespoke software, the lack of formalised methodologies for creating new component models and for interfacing third party component models, combined with the diverse range of models, levels of model abstraction and modelling languages necessary to represent standard components. Furthermore, many approaches only provide for a fixed predetermined structure or assembly size, and where standard components are considered abstractions of the representations must be created by the user within the modelling environment, which is a serious limitation.

One of the fundamental issues to be addressed by a modelling approach is the ability to consider a system of individual components as a whole, whilst providing for the specification and selection of components from third party representations. This follows the hypothesis that the electronic representations for standard components can be manipulated in such a manner so as to enable the performance of mechanical systems to be represented, and is addressed through the development of an integrated modelling approach. The incorporation of this modelling approach within a software environment addresses the second hypothesis of this work, which proposes that the modelling approach can be implemented in a computer based support tool to enable the representation of topology and performance for conceptual systems of standard components.

This chapter addresses this second hypothesis and describes the software components of a computer based support tool that incorporates the overall modelling approach developed in this work. The approach enables the representation of a mechanical system through the integration and manipulation of the various representations for individual mechanical components. And in particular, deals with the inclusion of electronic representations for standard mechanical components. These representations include electronic catalogues, CAD systems, spreadsheets and formalized numerical procedures for component design; all of which are software entities in their own right. It also provides for representations configured in proprietary software environments such as Microsoft Excel and Access, and programming languages such as BASIC and C.

The following sections discuss the implementation of the various aspects of the overall modelling approach, develop in chapters 4, 5, 6 and 7, within a software environment. The construction of a system model is described and a case study is used to illustrate the approach. The key aspects of the software environment include; system representation, system resolution, the handling of interactions, integrating electronic representations, data arbitration and compatibility analysis. Their functions within the overall modelling approach are depicted in figure 8.1.

8.1 System representation

In engineering design, systems are often considered to be a hierarchy of assemblies, sub-assemblies and components (Ulrich & Eppinger, 2000; Kusiak & Larson, 1995). The definition of these terms and the extent to which they encompass elements are held largely in the minds of the individual (Culley & Theobald, 1997). Whilst the designer may distinguish between assemblies and subassemblies, the modelling environment considers the complete machine system and each of its included elements, as shown in figure 8.2. Where this machine system encompasses all the primary mechanical components. For the purpose of the case dealt with, *primary elements* are those mechanical components that provide the overall transmission requirements of the system. *Secondary elements* are designed post selection of the primary components and may include housings and casings.

In order to achieve the necessary level of flexibility and completeness in the system representation, no predefined structure or order of arrangement is imposed on the designer. The form of system representation is shown in figure 4.2. This representation captures the relative arrangement and connectivity between components in the machine system. In the modelling environment this connectivity is extrapolated from the schematic layout constructed by the designer, an example of which is shown in figure 8.3.

Once a design schematic or layout of the machine system has been configured, similar to figure 8.4 a system template can be generated. This template describes the order of connectivity between each element in the system schematic. In addition to this, individual elements and the extents of their connections are evaluated, and categorised as either a unitary, binary or core element. A system template generated for the example in figure 8.3 is depicted in figure 8.4.

8.2 System resolution

In order to achieve the requirements for data propagation within the system model, the modelling environment implements the strategy described in section 4.3. The resolution cycle commences at the unitary elements, sequentially resolving binary elements until a core element is reached. Following this, resolution flags are reset and a second phase of resolution initiates from each core element through the binary elements, until either a unitary element, core element or a previously resolved element is reached. These resolution episodes are derived from the system template, which creates a command list for the modeller. This list contains information on the type of electronic representation, the location of the electronic representation, the type of component(s) to which it is connected and an identifier for the relative connection(s). This identifier relates to the 'local blackboard' which is associated with every connection in the system model. An example command list is shown in figure 8.5.

The primary function of the resolution procedure is to ensure that the design data necessary for the effective execution of each electronic representations is available. The majority of this data will be imposed by interactions with connected components and in the modelling environment the content and extent of this data is governed by the protocol for handling interactions.

8.3 A protocol for handling interactions

A mechanical system may consist of any number of components connected in a variety of configurations. This variety and diversity are the fundamental issues addressed by a strategy for system resolution and a protocol for handling interactions. These two components of the modelling approach dictate the order and relative timing of data propagation and the amount of data that is propagated. During the embodiment of a system the designer must evaluate constrained parameters between connected components, these include speeds, loadings, torques and geometric attributes. In order to automate this process, a protocol that communicates all necessary parameters between related components was developed in section 4.2. These parameters provide the basis on which selection data for individual components can be generated and the overall system performance evaluated.

The investigation of mechanical component attributes undertaken in chapter 4, reveals that component attributes may be classified into three-tiers; global, local and intrinsic attributes. These tiers or classes differentiate component attributes according to their method of formulation, and relates their dependency on the system, connected components and intrinsic properties of the component itself. Attributes in the global, local and intrinsic classes are therefore dependent on system data, data from coupled elements, or data that is particular to the component type respectively.

Global attributes, such as life and working environment, can be explicitly defined at a system level. This is achieved by the creation of a global data array that comprises system attributes entered by the designer, depicted in figure 8.6. In contrast, intrinsic attributes depend on data that is particular to the considered component and therefore intrinsic attributes need only be available at a component level. These requirements are met by the incorporation of data fields for each system element, shown in figure 8.7. These data fields are held in a system array which comprises an individual identifier for each component, all the attributes and their corresponding values, and a ranking for each attribute which is used for data arbitration, discussed in section 8.4. In the case of local attributes, the vast number and types of engineering components and associated attributes, prevents the explicit definition of all possible component attributes. As a consequence of this, only a range of design parameters are implemented, which are sufficient to represent the interactions between connected components. When these parameters are used in conjunction with the global and intrinsic attributes, they enable either the explicit definition of, or the derivation of all primary selection attributes for mechanical components. These local attributes are held and conveyed by the system blackboards which are represented as a three-dimensional array, illustrated in figure 8.8. In addition to conveying the data values, the blackboards also carry an associated ranking for each parameter which is determined by the arbitration routines discussed in section 8.4.

8.4 Data arbitration

The arbitration of related attributes between connected components is essential so that conflicting requirements can be resolved during the embodiment of a system. Such conflicts may include the diameter of a portion of a shaft and the internal diameter of a bearing. The modelling environment provides for two levels of data arbitration. The first of these is at a system level. Constraints between elements are specified as logical conditions, which are evaluated concurrently during resolution. These conditions are generally expressed as numeric equalities, although string comparisons can be evaluated, but this is restricted to exact matching only. The constraints are specified by the designer and held in a system array, the interface for which is depicted in figure 8.9. This array contains the condition to be satisfied, the name of each attribute and the associated identifier for each component. The latter of which is assigned to each system element during model construction. This enables the attribute values to be identified within the system arrays and the specified constraint to be evaluated. This feature can be utilised for ensuring attributes such as lubrication types are consistent throughout a design. If the constraints are not satisfied the designer is notified and prompted to take action in order to resolve the conflict and satisfy the constraint.

The second level of data arbitration is implemented at an elemental level and occurs prior to the interrogation of each electronic representation. The strategy developed in this work, involves agents which undertake the independent resolution of conflicts between the desired attribute values for a component and the constraints imposed by connected components. Because this work aims to interface existing third party representations, agents cannot be incorporated within the representations themselves. As a consequence, virtual agents are used to arbitrate data prior to the execution of the third party representation. These virtual agents are separate software modules that provide the necessary levels of data arbitration for each electronic representation. In order to perform data arbitration the virtual agents compare the rankings assigned to component attributes and the rankings assigned to the constraints imposed by connected components. These rankings are incorporated in the system array for component data and the array structure for the blackboards, described in sections 8.2 and 8.3 respectively.

The inclusion of agent based arbitration means that constraints are registered and resolved automatically whilst ensuring that less important requirements and constraints on the system do not overwrite more important ones.

8.5 Compatibility analysis

In order to provide for compatibility analysis within the modelling approach a knowledge base was developed in section 6.5.3. This knowledge base explicitly defines compatible, incompatible and complementary component combinations. The knowledge base is represented in the modelling environment as a 'system array' which is accessed and updated by the designer each time a new electronic representation is interfaced with the system modeller. The software tool for editing the knowledge base is depicted in figure 8.10. This knowledge base provides two essential support functions. During model construction, the knowledge base can be recursively interrogated to generate a listing of all compatible component sequences, this is shown in figure 8.11. The current implementation limits this hierarchy to a three level listing, although this can be extended if required. The second function of the knowledge base is to evaluate component compatibility after the model has been resolved. This involves sequentially evaluating component connections to determine whether there are any incompatible combinations. This evaluation uses the system template, discussed in 8.2, and interrogates the knowledge base for each connection to determine whether the two components coupled by the connection are a compatible, complementary or incompatible.

8.6 Interfacing electronic representations

Electronic representations are for the purpose of this work, independent software entities that size or select a real engineering component. An essential part of the approach of this work is to provide for the incorporation of third party representations within the system approaches. These software entities can be bespoke codings, executables or application dependent software modules, such as spreadsheets or databases.

The strategy for interfacing electronic representations is depicted in figure 8.12 and utilises interprocess communication and the blackboard architecture implemented in the modelling environment. The content and structure of the blackboard architecture is described in detail in chapter 4. Three classes of data need to be communicated and include global attributes, local parameters and component attributes held in the element data fields. The first two classes are contained within the system blackboard structure and are each assigned individual identifiers shown in figure 8.13. The attributes held in the element data field also possess individual identifiers which are assigned to each element of the system during model construction. The communication of data between the modelling environment and the individual electronic representations is enabled through the Dynamic Data Exchange (DDE) convention implemented in the Microsoft Windows environment. This allows interprocess communication at runtime enabling independent software applications to exchange data. This communication involves the exchange of individual units of data, which are identified by their application and corresponding item, in this case the identifiers associated with each blackboard. This allows all the necessary attributes and parameters to be transferred to a third party electronic representation for remote interrogation.

Currently implemented within the modelling environment are software components for DDE transfer between the system environment and proprietary software applications such as Microsoft Excel and ACCESS, SolidWorks, and programs written in BASIC and C, as well as, an ASCII data file that carries the same level of data for interfacing applications that do not support the DDE convention. Research work is also being undertaken into developing methods to allow interprocess communication and remote interrogation of electronic catalogues (Allen *et al*, 2001). The generic structure of the software components is shown in figure 8.14. These components also encapsulate the data arbitration routines discussed in section 8.4. For the purpose of interprocess communication, these software components activate the representation, assimilate and arbitrate the data upon which the design or selection of the component will be based, exchange this data and invoke the necessary procedures or events. The results of the interrogation are captured and compiled by the system, following which the electronic representation is deactivated or closed.

8.7 Configuring a system model

The procedure for configuring a system model is illustrated in figure 8.15. The solution principle must firstly be determined by the designer, as illustrated in part (a) of figure 8.16. From this solution principle, a schematic can be constructed which represents the connectivity of the conceptual solution. Individual components, represented as icons, are placed on to the worksheet and linked to form the mechanical structure (i.e. the system configuration). Once completed, the appropriate electronic representation that governs the design and selection of each mechanical component must be chosen. This is undertaken for all but the core elements, for the case considered in figure 8.16 the core elements are the shaft components. In order to select a representation and determine the order of connections for a core component, knowledge describing all the connected component types must be available. In order to generate this knowledge, the system template is created prior to selecting the governing representation(s) for the core element(s). This template contains the necessary data to describe each connecting element and associated order of connectivity with the core element. The default order of connection follows the order in which the model was constructed. Although this order or arrangement of connections can be changed by the designer and is achieved by reordering the system template. The interface for this is depicted in figure 8.17.

Following the assignment of appropriate representations for each component, system constraints must be set and the performance requirements for the system and individual components specified. Constraints are set between attributes of related components, this relation does not have to be a direct coupling, it could be between two shafts which are connected by other elements. Performance attributes are specified in either the global data field or the attribute data fields for the appropriate components. Once complete, the system model may be resolved and a system of real, compatible mechanical components which satisfy the desired performance requirements are generated.

8.8 Case example

The case example shown in figure 8.16 is the drive-system for a film over wrapper. These machines are commonplace in industries where secondary or tertiary packaging is required for consumables such as CDs, teabags or cigarettes. The conceptual design sketch shown in part (a) of figure 8.16 can be decomposed into a design schematic or connectivity diagram shown in part (b) of figure 8.16. The two cam and follower assemblies are represented as outputs or system loads, whilst the electric motor is considered to be a system input, for which the speed and power are specified. The additional layout constraints for the shaft center distances are embodied in the system model by specifying the center distances for each chain drive in their respective representations. In the construction of a system model, a schematic is firstly constructed by placing iconic representations onto the worksheet, the connectivity model is then created by linking these icons. Once complete, the various representations that govern each mechanical component must be selected. The bearings are selected from a catalogue of cylindrical roller bearings (SKF Limited, 1998); keyways are specified using design algorithms for keyway design programmed in BASIC; chain drives are designed utilizing a third party design tool, whilst the shaft algorithms are written in C and the loading elements represented in the Microsoft Excel environment. An overview of the various electronic representations is included in the Appendix. In the case study described, performance data for the centre distances of the chain drives, the maximum loading case for the cams and the power of the motor are entered. The system model is resolved and a set of components is determined. The resulting embodied solution can also be represented as a set of basic solids, shown in part (c) of figure 8.16. These solids are extracted post resolution, from the geometric data packets contained in the system's local blackboard structure, described in section 4.2. This provides data for the centroid and spatial envelope of each component. The production of these simple solids provides a visual verification of the embodied system, and represents the sizes and arrangement of the components.

The embodied solution generated within the modelling environment meets all the desired performance criteria as well as the additional layout constraints. The resolved system contains real components selected from third party catalogues or designed through standard procedures, and may be procured exactly as specified in the system model. The resultant system of components are fully compatible and physically connectible, their spatial arrangement is captured in the arrangement of solids shown in part (c) of figure 8.16. Although component intricacies are not shown, the level of geometric representation is sufficient to provide data on which decisions such as housings, spatial occupancy and fixtures may be taken.

8.9 Concluding remarks

During the transformation of a design concept to an embodied solution a lot of time is expended by the designer as key decisions are made. These decisions pertain to the selection and viability of design configurations and the selection of mechanical components to achieve both performance and spatial requirements. The traditional embodiment process demands that the designer manipulate related attributes and dependencies between components, which is a time consuming and analytically intensive task. This chapter discusses a computer based support tool which enables the representation of topology and performance of conceptual systems of standard components. The creation of a support tool for the embodiment of conceptual systems from standard components addresses the second hypothesis of this work.

The modelling environment developed in this work deals with third party electronic representations, which enables the actual supplier or manufacturers data to be used and more fully informed decisions to be taken by the designer. The innovative aspect of the modelling environment described, is the ability to represent a mechanical system utilising third party electronic representations. This is achieved through the implementation of a strategy for transferring data within the system model and between electronic representations, as well as standard software procedures for interfacing component representations based on the Windows Dynamic Data Exchange protocol. These features enable the construction of a system model that considers an assembly of components and their associated electronic representations as a whole. Furthermore, procedures for arbitrating conflicting data between system elements, ensures that a fully compatible system of mechanical components is automatically generated. The modelling environment provides a tool for the embodiment of design concepts with standard mechanical components. This embodiment is driven by both performance criteria and geometric constraints. The system environment allows for the integration of various electronic representations such as electronic catalogues, advanced numerical methods and CAD models, enabling real components and their associated selection procedures to be considered.

The automation of the embodiment process enables the designer to consider many more configurations, component types and sizes within a shorter time. If only because of the removal of time-consuming data manipulation tasks. This enables the designer to develop a better performing solution, which might not have been obtainable if the more traditional, manual procedures for embodiment were used. Another benefit, is that design models can be saved and reused. This enables standard design models to be scaled and refined as required, perhaps for different performance requirements, a changed supplier or different component specification.

The modelling approach also provides a platform for optimisation which utilises real[1] components and therefore a real solution space. The issues associated with the optimisation of systems of standard components are discussed in chapter 10 and in particular the ability to consider mass, spatial occupancy and cost as optimisation goals are addressed. In addition to this chapter 9 discusses the limited availability of costing information within third party representations and proposes a number of cost forecasting algorithms to enable the generation of cost information for system models.

[1] Real components are those elements which may already exist, have been previously used and have predetermined or predefined properties. In many cases, these are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified.

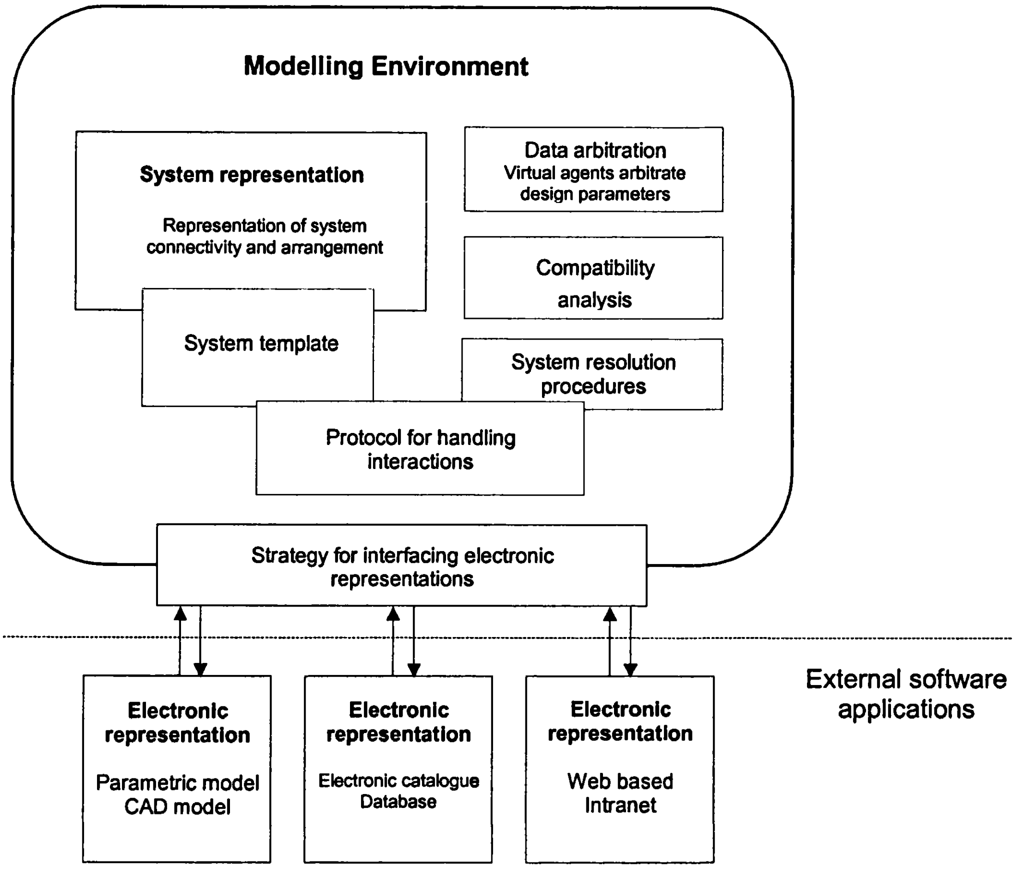


Figure 8.1 – The key aspects of the integrated modelling environment

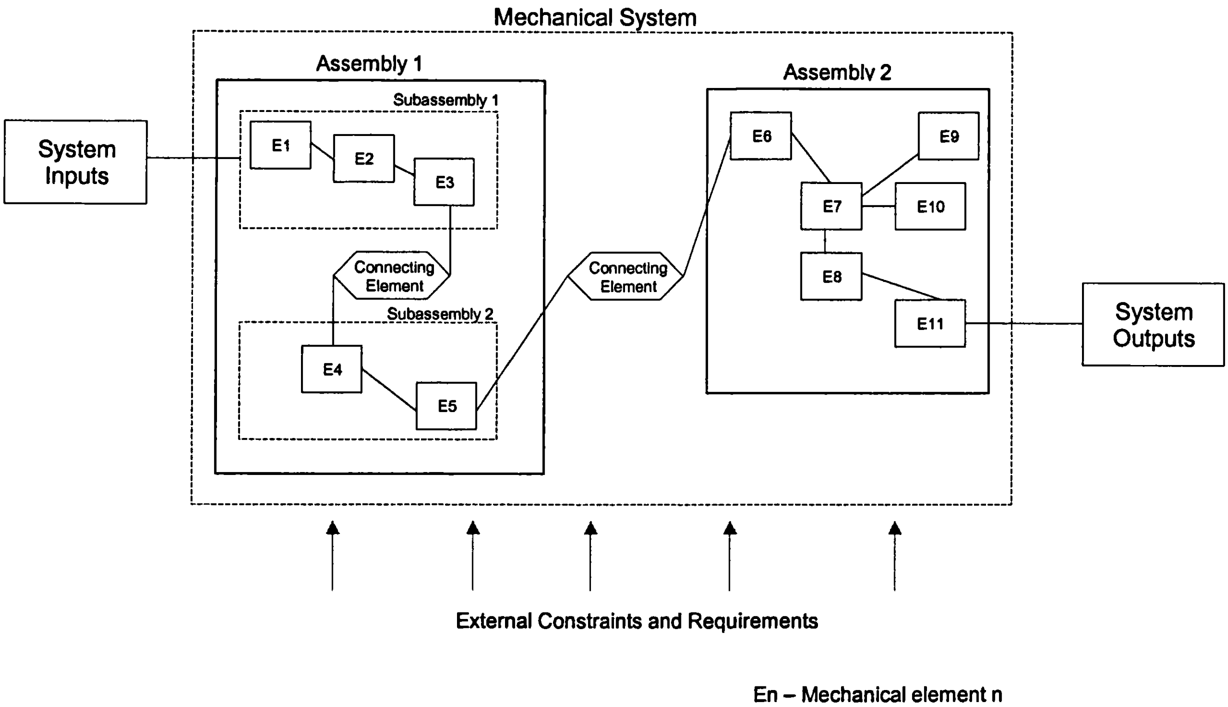


Figure 8.2 – Decomposition of a mechanical system into assemblies, subassemblies, components and their connectivity

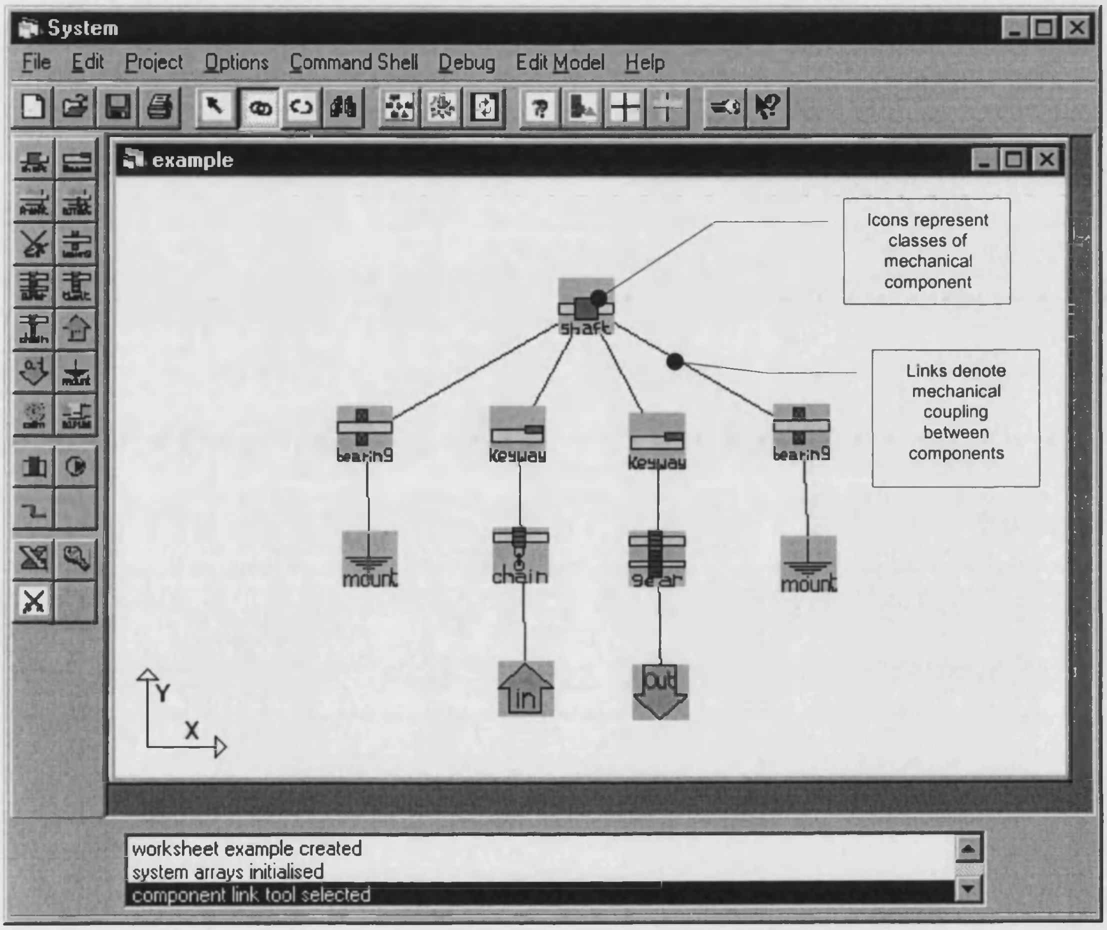


Figure 8.3 – Example schematic constructed in the integrated modelling environment

Component type (c1)	Control index	Element type	X location (c1:e1)	Y location (c1:e1)	Link index	X (e1)	Y (e1)	Link type	X (e2)	Y (e2)	Link index	X location (c2:e2)	Y location (c2:e2)	Element type	Control index	Component type (c2)
Shaft	1	H	3765	840	1	4005	1080	HB	2160	2115	1	1920	1875	B	5	Bearing
Bearing	5	B	1875	1920	2	2115	2160	BU	3225	2160	2	2985	1920	U	10	Mount
Shaft	1	H	3765	840	3	4005	1080	HB	2160	3420	3	1920	3180	B	2	Key
Key	2	B	3180	1920	4	3420	2160	BB	3195	3450	4	2955	3210	B	4	Chain
Chain	4	B	3210	2955	5	3450	3195	BU	4335	3495	5	4095	3255	U	3	Input
Shaft	1	H	3765	840	6	4005	1080	HB	2220	4605	6	1980	4365	B	7	Key
Key	7	B	4365	1980	7	4605	2220	BB	3225	4605	7	2985	4365	B	8	Gear
Gear	8	B	4365	2985	8	4605	3225	BU	4365	4635	8	4125	4395	U	9	Output
Shaft	1	H	3765	840	9	4005	1080	HB	2145	5820	9	1905	5580	B	6	Bearing
Bearing	6	B	5580	1905	10	5820	2145	BU	3270	5880	10	3030	5640	U	11	Mount

Figure 8.4 – System template representing connectivity within the model

Component	File path	Control index	Link index	Control index	Connected component
Mount	C:\MASCtool\vbmodels\mount.exe	10	2	5	Bearing
Bearing	C:\MASCtool\accessmodels\bearing.mdb	5	1	1	Shaft
Input	C:\MASCtool\excelmodels\input.xls	3	5	4	Chain
Chain	C:\MASCtool\vbmodels\chain.exe	4	4	2	Key
Key	C:\MASCtool\vbmodels\key.exe	2	3	1	Shaft
Output	C:\MASCtool\excelmodels\output.xls	9	8	8	Gear
Gear	C:\MASCtool\excelmodels\gears.xls	8	7	7	Key
Key	C:\MASCtool\vbmodels\key.exe	7	6	1	Shaft
Mount	C:\MASCtool\vbmodels\mount.exe	11	10	6	Bearing
Bearing	C:\MASCtool\accessmodels\bearing.mdb	6	9	1	Shaft

Part (a) Inward phase of resolution

Component	File path	Control index	Link index	Control index	Connected component
Shaft	C:\MASCtool\cmodels\shaft.mac	1	1	5	Bearing
Bearing	C:\MASCtool\accessmodels\bearing.mdb	5	2	10	Mount
Key	C:\MASCtool\vbmodels\key.exe	2	4	4	Chain
Chain	C:\MASCtool\vbmodels\chain.exe	4	5	3	Input
Key	C:\MASCtool\vbmodels\key.exe	7	7	8	Gear
Gear	C:\MASCtool\excelmodels\gears.xls	8	8	9	Output
Bearing	C:\MASCtool\accessmodels\bearing.mdb	6	10	11	Mount

Part (b) Outward phase of resolution

Figure 8.5 – Command list for inward and outward phases of resolution

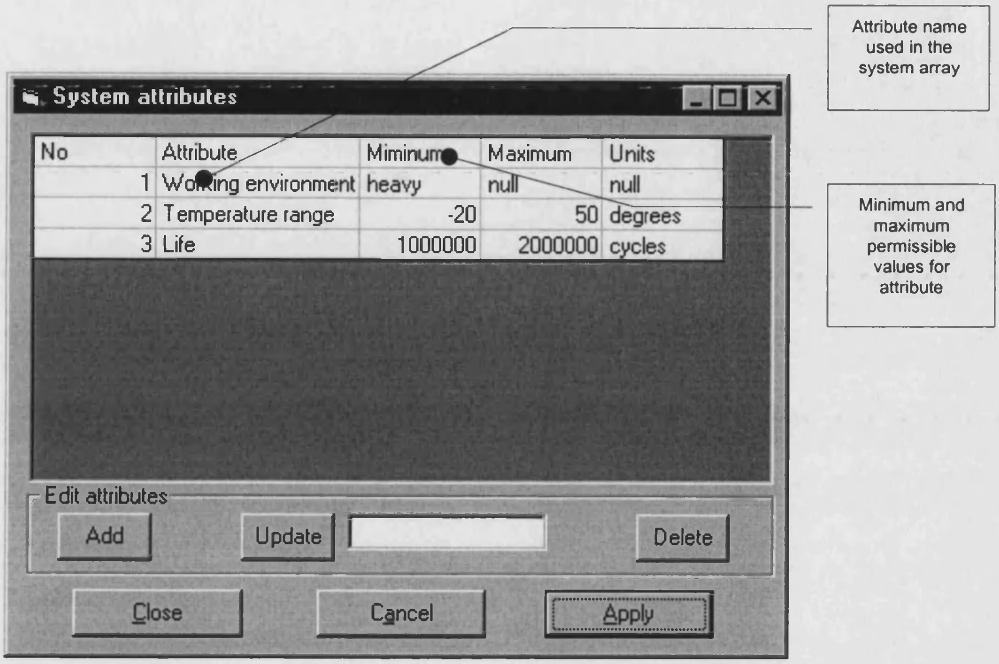


Figure 8.6 – Specification of system level attributes

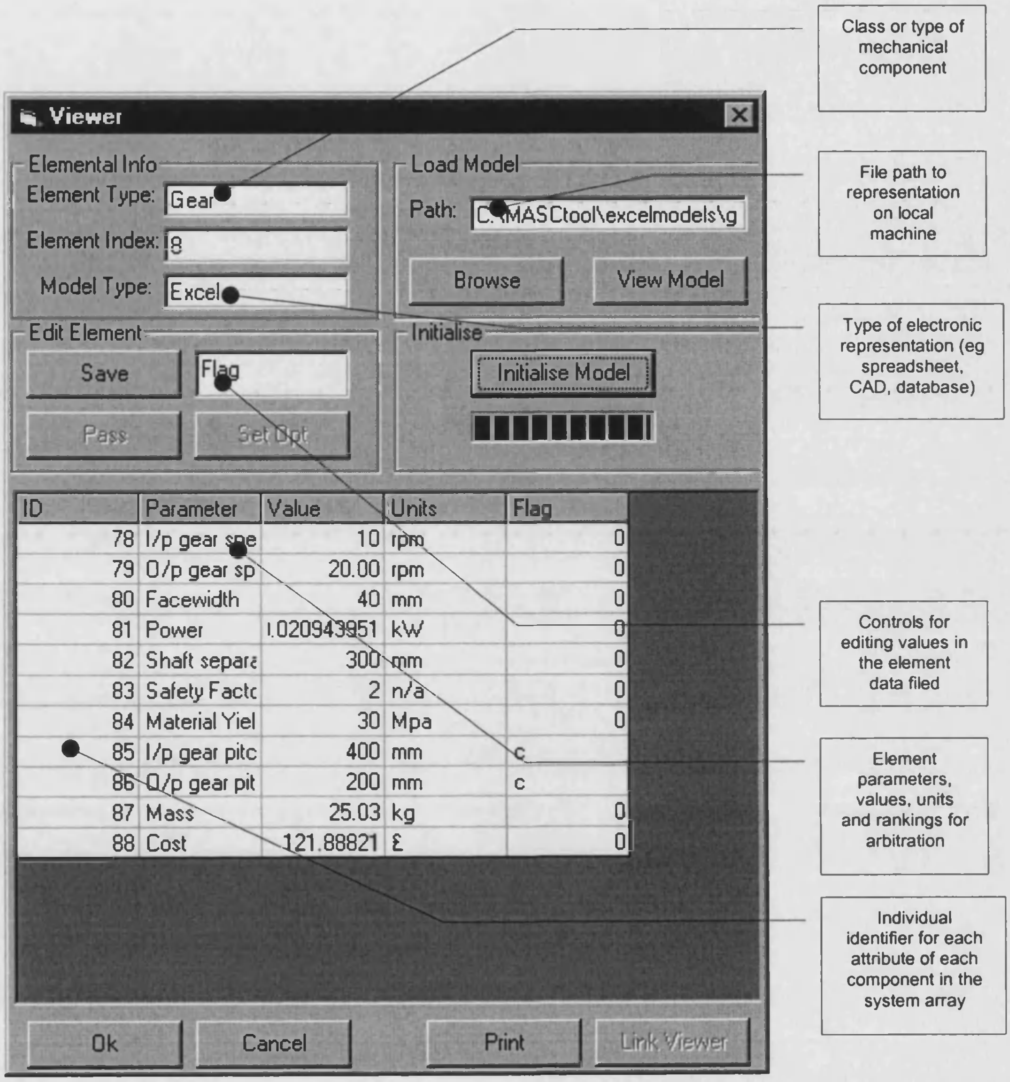


Figure 8.7 – Data field for a gear component

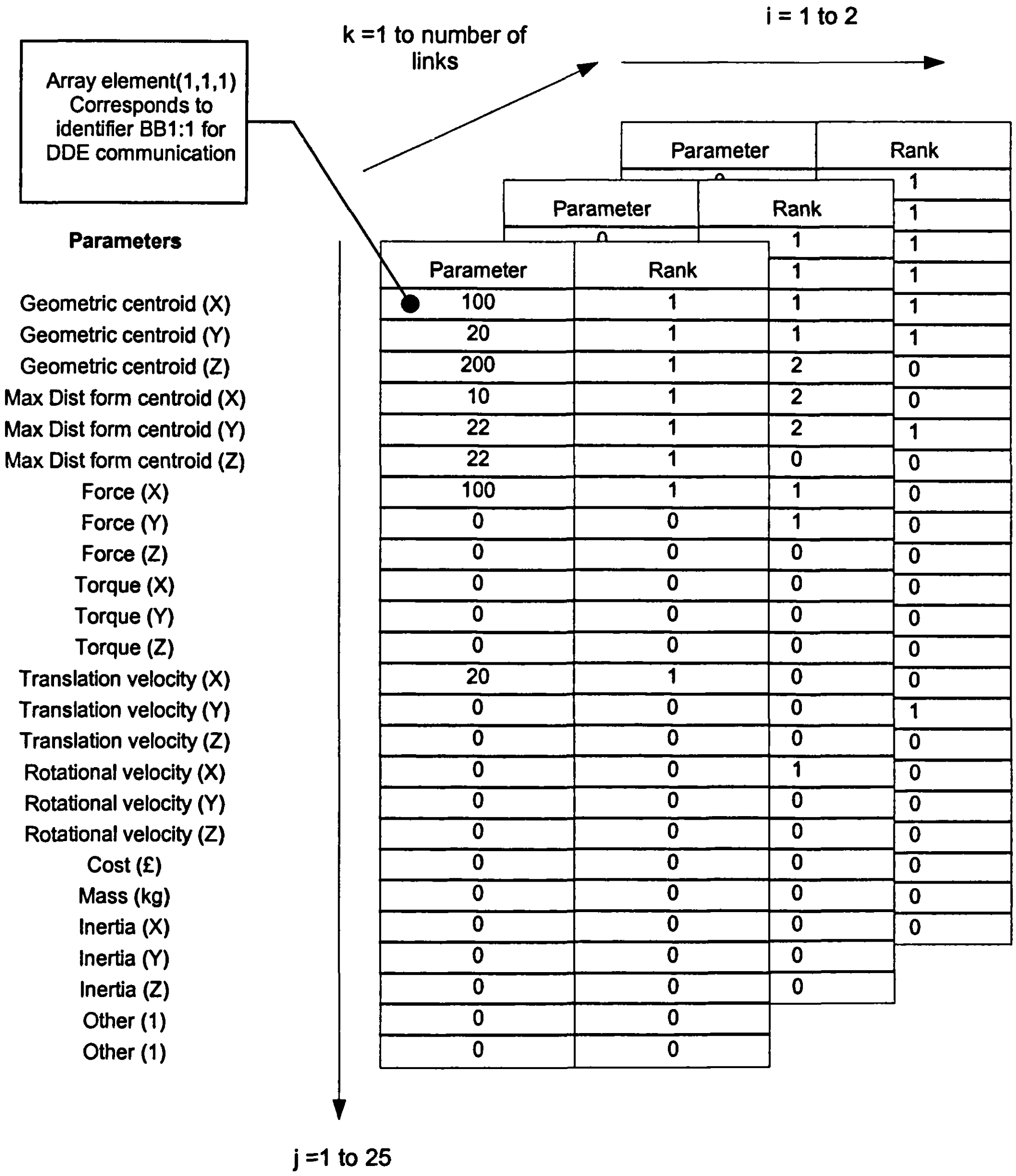


Figure 8.8 – Three-dimensional system array for the local blackboard structure

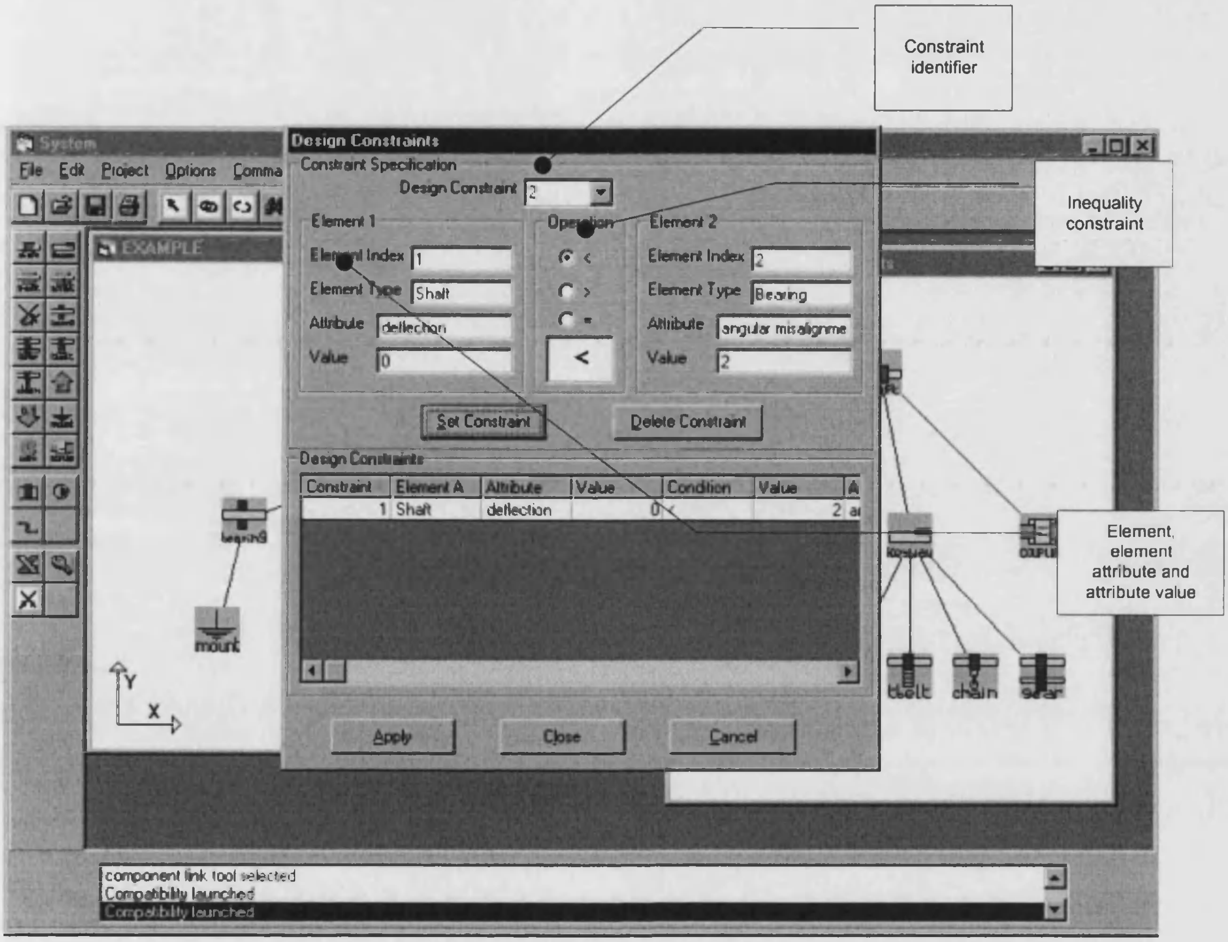


Figure 8.9 – Specification of compatibility constraints between related attributes

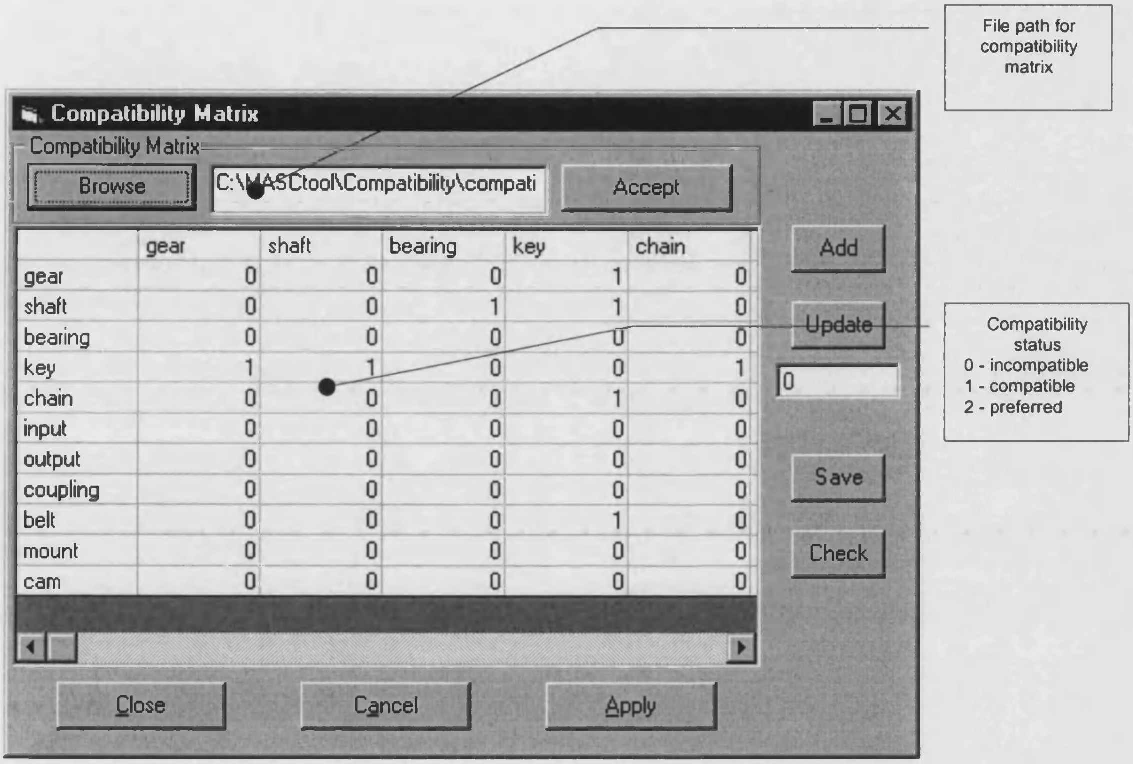


Figure 8.10 – Compatibility knowledge base

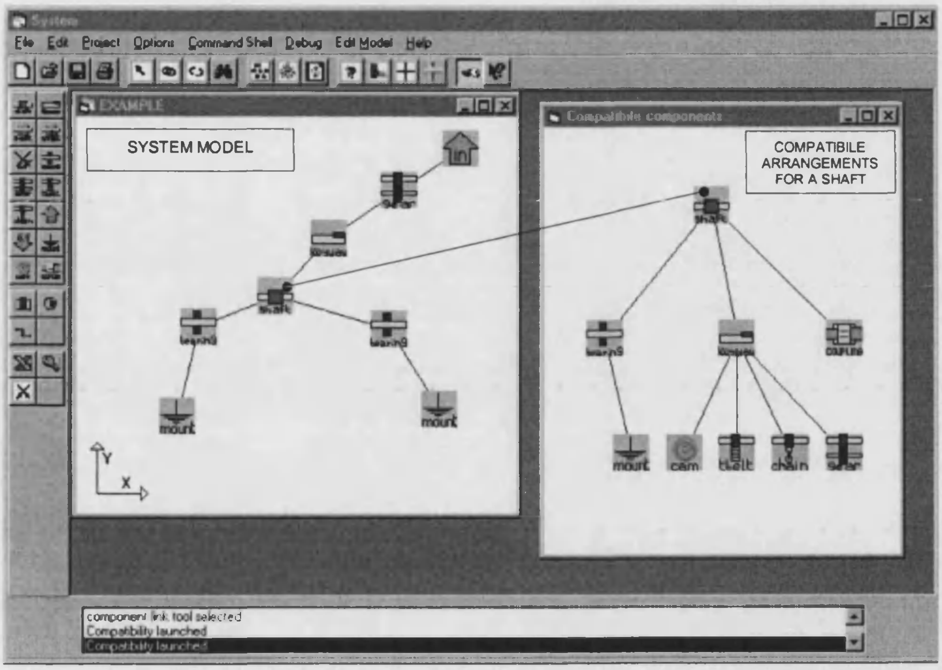


Figure 8.11 – Compatibility analysis during model construction

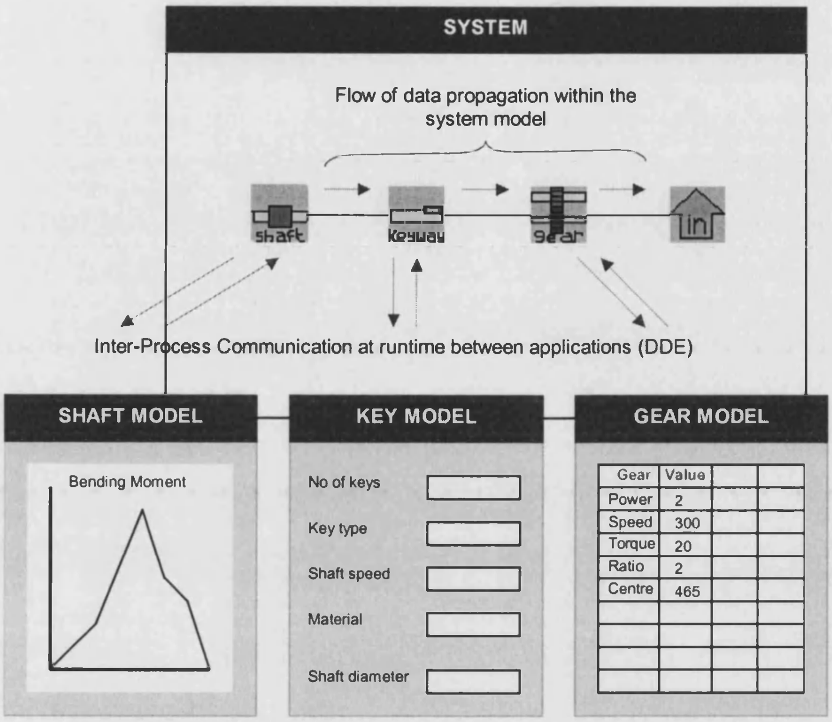


Figure 8.12 – A systems approach to interfacing electronic representations

Whiteboard data packet	Whiteboard data fields (read status only)	Identifier Assembly No : Data Element No.
Global	Life (hrs)	WB1:1
	Reliability	WB1:2
	Safety factor	WB1:3
	Lubrication	WB1:4
	Temperature	WB1:5
	Humidity	WB1:6
	Other1	WB1:7
	Other2	WB1:8

Blackboard data packet	Blackboard data fields (read & write status)	Identifier Board No : Data Element
Geometric/form	Geometric centroid (X)	BB1:1
	Geometric centroid (Y)	BB1:2
	Geometric centroid (Z)	BB1:3
	Max Dist form centroid (X)	BB1:4
	Max Dist form centroid (Y)	BB1:5
	Max Dist form centroid (Z)	BB1:6
Physical	Force (X)	BB1:7
	Force (Y)	BB1:8
	Force (Z)	BB1:9
	Torque (X)	BB1:10
	Torque (Y)	BB1:11
	Torque (Z)	BB1:12
Motion	Translation velocity (X)	BB1:13
	Translation velocity (Y)	BB1:14
	Translation velocity (Z)	BB1:15
	Rotational velocity (X)	BB1:16
	Rotational velocity (Y)	BB1:17
	Rotational velocity (Z)	BB1:18
Auxiliary	Cost (£)	BB1:19
	Mass (kg)	BB1:20
	Inertia (X)	BB1:21
	Inertia (Y)	BB1:22
	Inertia (Z)	BB1:23
	Other (1)	BB1:24
	Other (1)	BB1:25

Figure 8.13 – Example data structures and identifiers for communication of data within the system model

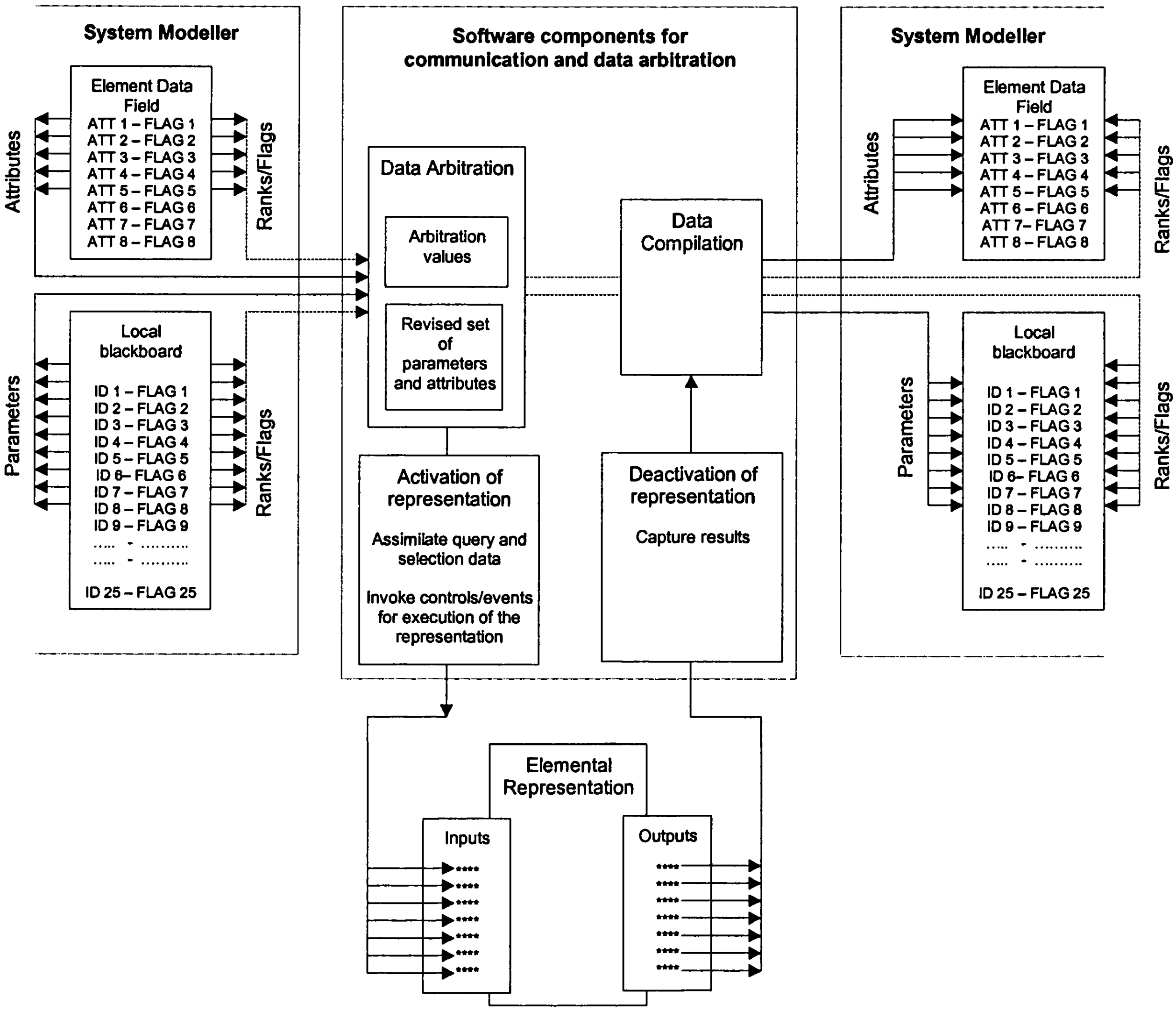


Figure 8.14 – A generic structure for data input, arbitration, assimilation and interrogation of a component representation

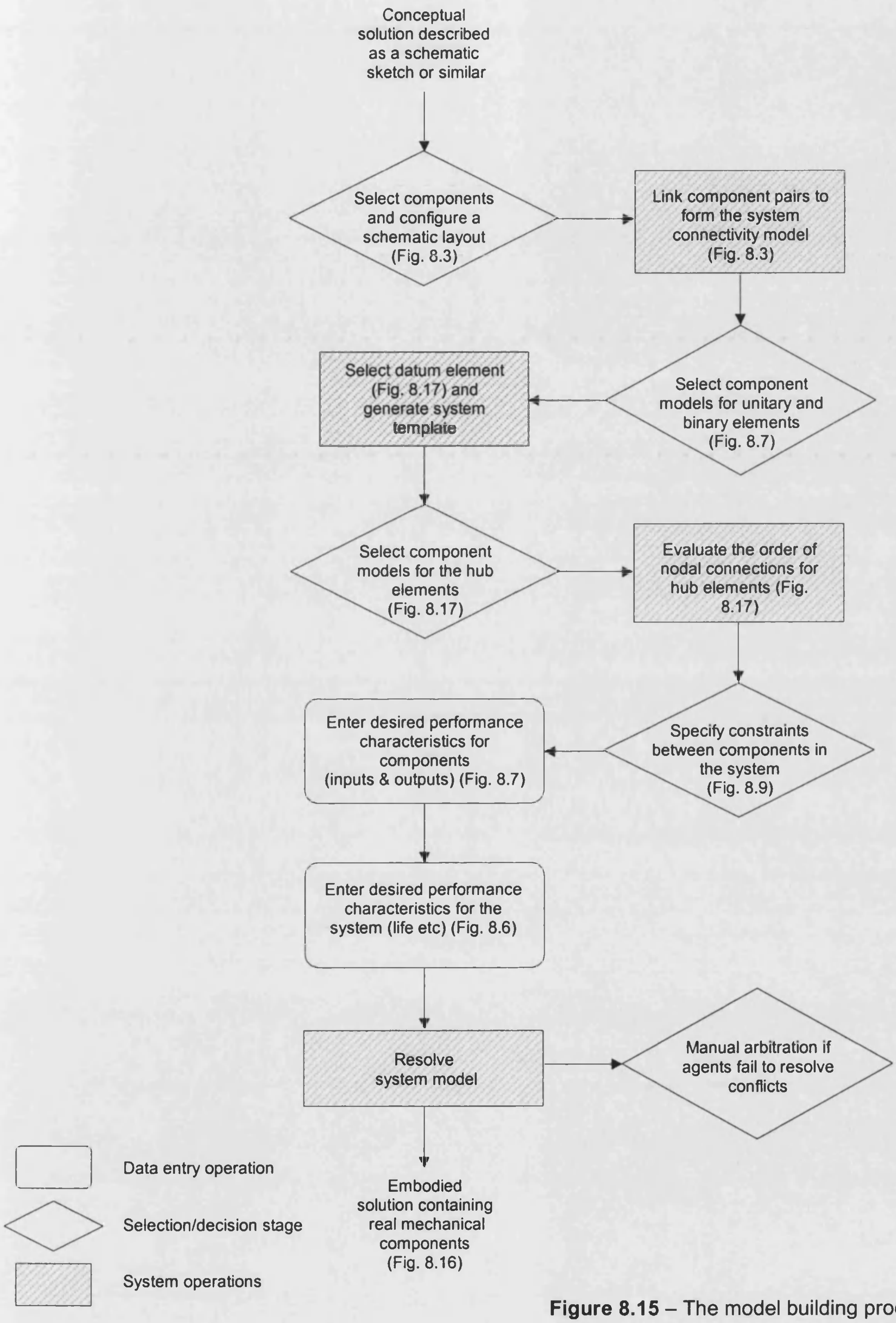


Figure 8.15 – The model building process

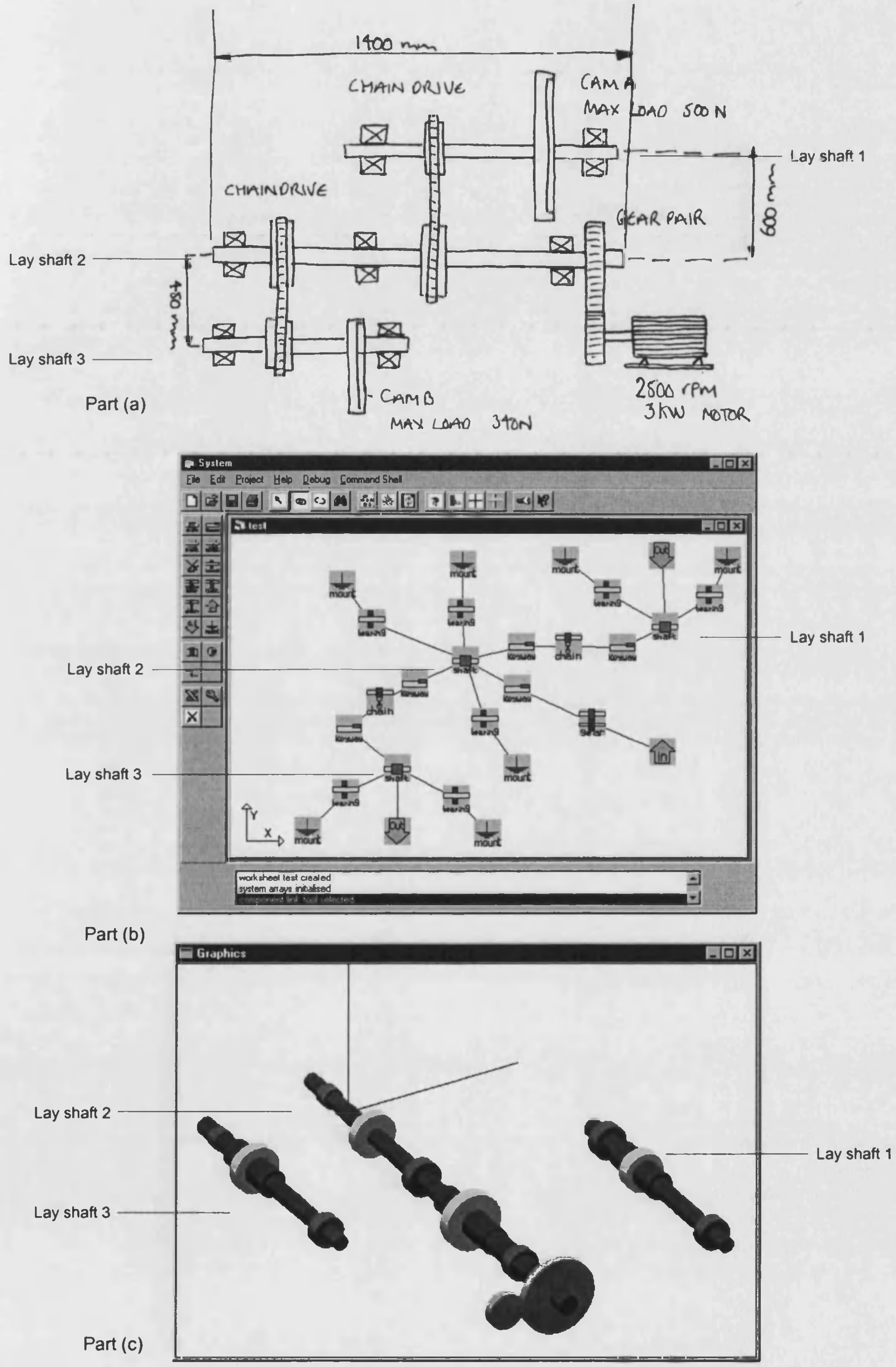


Figure 8.16 – Modelling episode for embodiment of a drive-system for an over wrapper

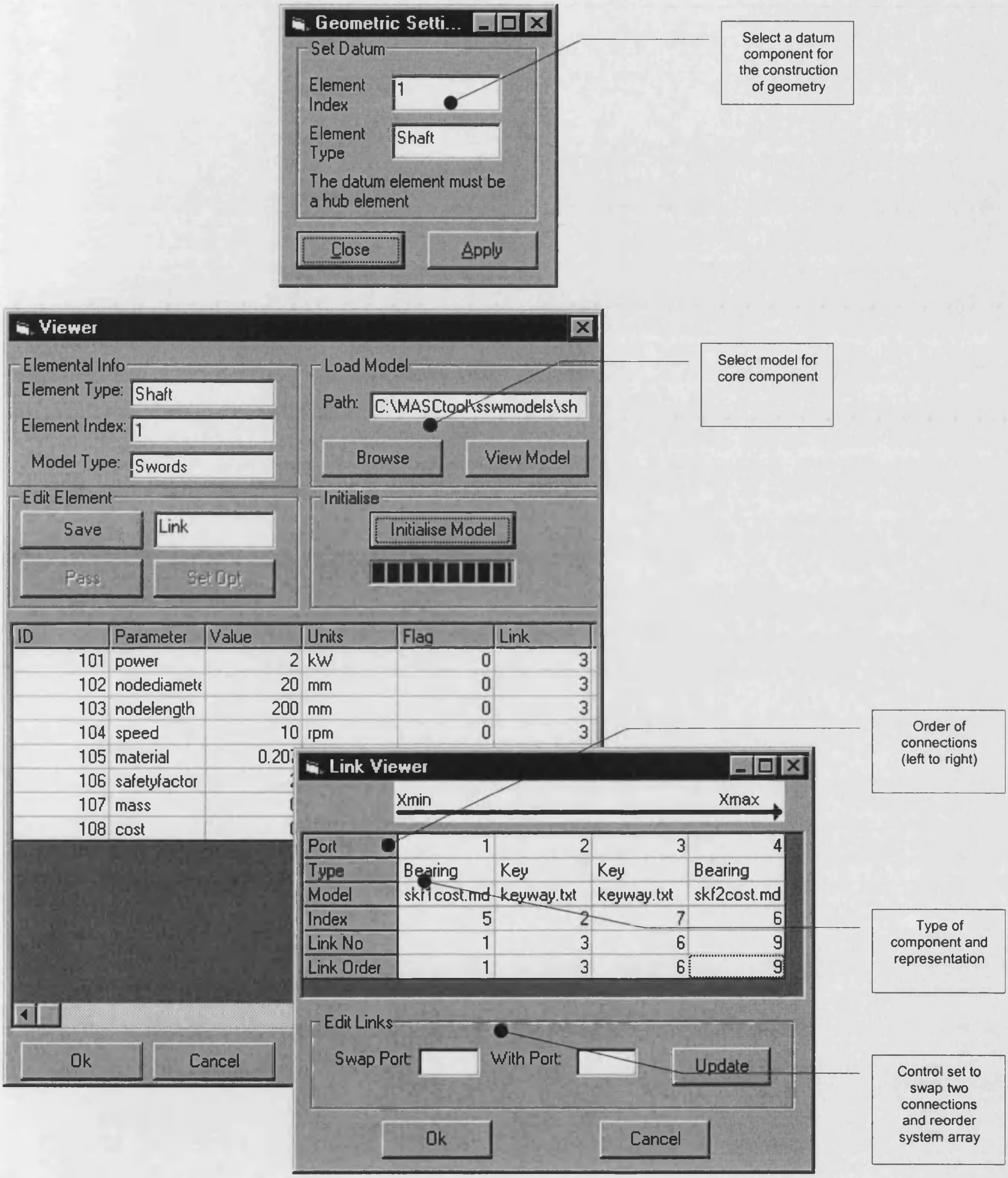


Figure 8.17 – Reconfiguration of the order of connections for core elements

Chapter 9

Cost estimation for standard components and systems

The previous chapters discuss the development of a new modelling approach and its incorporation into a computer based support tool. These chapters address the first and second hypotheses of this work. The third hypothesis proposes that the approach developed can be extended to enable configuration, embodiment and optimisation of engineering systems from standard components. The issues of embodiment and configuration are dealt with in the development of the overall modelling approach, and the issues associated with optimisation are dealt with in this chapter and chapter 10. In particular, the previous chapter highlights the need to design systems in consideration of the overall performance capabilities, spatial occupancy, overall mass and total cost. The approach developed so far in this work, provides for the goals associated with performance, mass and spatial occupancy. However, the inclusion of cost as a goal function is frustrated by a number of issues. These issues are discussed in this chapter and a strategy is developed to enable the consideration of cost in the modelling approach.

One of the most difficult tasks undertaken by the designer is to evaluate the cost of a design. This is a very important consideration in the design process, especially when the designer is trying to choose between alternatives, or optimising a particular solution. The designer must develop a design in order to fulfil the required performance characteristics as well as provide the appropriate quality at minimal cost. In the development of new products and systems, the pursuit of reduced development time and costs demand that decisions are taken at the earliest phase in the design process (Ullman, 1992). This is because a retrospective drive to reduce a particular aspect of a design is far more costly than undertaking the appropriate consideration and analysis earlier in the process (Pahl & Beitz, 1996). It is therefore, the earlier phases of the design process; the selection of a concept and the transformation of the concept into an embodied solution, which have the greatest influence on the cost of the final designed artefact. In fact, it is widely accepted that upwards of 80 percent of final cost is obligated when the solution principle has been selected and its embodiment completed (Pahl & Beitz, 1996).

In order for a product to be truly competitive, its price must be favourably comparable with like products of similar quality, performance and functionality. It is therefore desirable to generate an indication of cost as early as is practicable (Mansour, 1999; Samuel & Weir, 1999). However, more often than not early decision making implies that decisions must be taken at a high level (McMahon & Browne, 1993). This means that costing information is derived from brief descriptions of the type of component or from relative costing between component types. The accuracy and reliability of the information generated through this approach is not sufficient to support the designer in making the full range of necessary decisions. These decisions include considerations of component types, sizes, combinations and also the choice of supplier.

To improve the accuracy and reliability of costing information, real fully specified components must be considered, and more representative cost values used. Although some CAD systems aim to provide for real components and their associated details, much of the information relates to attributes such as mass and leading dimensions, rather than cost. This is because accurate costs are increasing only obtainable during procurement from suppliers. This complication has arisen as a result of competition between suppliers, and their desire to deliver managed services to their customers (Culley & Webber, 1992). Many suppliers now utilise advanced data management systems to improve searching and provide up-to-date cost for individual components, as well as to manage their stocks, production and ordering at an enterprise level. These systems also yield significant advantages for the designer, such as the elimination of searching and the provision of accurate and competitive costs with associated reductions. However, they do tend to prevent open access to component information and in particular cost. This can frustrate the designer's ability to perform some important design tasks. This is often the case during the early stages of the design process where a designer may wish to compare many design configurations, component combinations and suppliers to assess relative costs. As a consequence, the designer requires access to cost information for the full range of component types and sizes provided by a supplier. If access to complete information is not provided then some form of forecasting or modelling is required so that cost information can be generated.

For the conceptual to embodiment stage of the design process, the majority of cost forecasting and selection decisions are relative, which on the assumption that costs are not discounted, provides a fair basis on which the most economical solution can be chosen. However, to effectively compare design alternatives the designer really needs the ability to forecast costs for the complete system. This usually involves the decomposition of the artefact into individual elements or components, similar to a bill of materials. In general, two categories of variables that drive the cost of individual components need to be considered. These are variables that describe the product and variables that describe the production environment. For third party components and in particular standard components only variables which describe performance capabilities and physical characteristics are considered, whilst for manufactured components it may be necessary to consider both variables in order to generate a truly representative costing model.

This chapter deals with the incorporation of a variety of costing techniques for the three classes of engineering component: standard selected, standard designed and bespoke designed within a systems modelling approach. This cost estimation is made from the performance attributes of individual components and represents the overall cost (of procurement) for the system. Cost forecasting techniques are reviewed and their limitations discussed with respect to systems of standard components. Following which, methods to incorporate costing information into component selection procedures are developed. These methods are applied to a number of component representations within the modelling environment and a case study is used to demonstrate the approach.

9.1 Cost forecasting in mechanical engineering

Considerable work has been undertaken in the area of costing and in particular for manufacturing and machining (Roztocki & LaScola, 1999; Aldrich, 1995; Stirk *et al*, 1998; Yang *et al*, 1998), whilst only limited research has been undertaken in the area of mechanical design and configuration from standard components. Figure 2.8 shows the results of a recent study, which reveals that standard components can provide between 50-80 percent of the mechanical elements in a design and therefore contribute, perhaps more significantly than manufactured elements, to the overall cost of the system.

A method for costing individual components is proposed by French (1990), who describes function costing techniques. These provide a technique for estimating relative costs of a particular component from the specification of its function (French, 1990). However, such techniques are complicated where the function of a component is not clearly defined, multiple functions are to be considered, or where a complete system is considered. For more general costing approaches Pahl and Beitz (1996) discuss methods such as similarity, relative costing and material costing, but these tend to be more applicable to manufactured components. More recent work deals with activity based costing. This approach assigns the cost of resources to the activities necessary to produce a particular component; these activities include all aspects of production and also services such as administration and distribution (Brimson, 1998). Whilst these may be accurate accounting tools, such procedures may not be truly useful to the designer until after the product has been designed in some detail. Some of the most recent work discusses feature-based costing (Brimson, 1998; Leibl *et al*, 1999). This approach aims to assign costs to the features of a component; these costs include the necessary activities for manufacture and production. Nevertheless, even this leading technique focuses on developing costing techniques for manufactured components rather than standard components.

Other techniques for the cost estimation of mechanical components include the evaluation of previous order systems and procurement records for the same or similar components (Ullrich & Eppinger, 2000). This produces accurate data for previously purchased components, which is certainly of use where companies standardise their utilisation of third party components. However, where new component sizes, manufacturers, types and configurations are considered relevant cost information is still not available. In such cases, heuristics or cost forecasts are often utilised. These provide for approximate component costs, often derived through statistical models obtained from sample sets of data. It is in this important area where there is a deficiency of formal or prescribed modelling procedures for cost forecasting.

In other disciplines, work has been undertaken on the parametric costing of components, where cost models are driven by the specification of the product (Hajare, 1998). Methods for dealing with the cost estimation of different component configurations have been developed for bare board manufacture in the electrical industry. Selling prices are extremely competitive and profit is determined by making effective use of cost optimised mixes of components (Robinson, 1991). Arguably, it is a combination of these two approaches; parametric costing for individual components and system costing for different configurations of components, which the designer requires. The development of formal methods that provide for these approaches would enable the designer to consider the cost implications of various configurations or design alternatives as early as possible during the design process.

The development and utilisation of generic costing techniques for mechanical systems is frustrated by a number of factors previously discussed, but particularly by the different classes of engineering component. These include standard selected, standard designed and bespoke designed components (Culley *et al*, 1997).

- Standard selected components are those elements that will ultimately be selected from a range of component sizes held by third party suppliers.
- Standard designed components maybe fully specified through practiced or established design procedures, although they may be restricted to certain standard size ranges.

- Bespoke elements are one-off components tailored to meet the specific requirements of a particular application. There maybe some formalized rules for their specification, however, much depends on the experience and knowledge of the designer as well as the application.

Each class of engineering component and their governing models possess varying levels of availability of information which can be used for the determination of cost, discussed in chapter 7. This situation is exacerbated by the various types of representation for each class of component. These range from electronic catalogues and spreadsheets to numerical procedures and CAD models. The approach of this work is to develop cost models that can be constructed from the selection data produced by these various representations. For the purpose of this work, the cost models derived, aim to predict the list price (cost of procurement) for a particular component or the cost of manufacture, for use in an overall system representation. Each costing technique is described in detail and applied to a number of mechanical components. These cost models are then introduced into the system modelling environment, enabling costing information to be generated for various mechanical configurations.

9.2 Classes of mechanical components and cost forecasting

Prior to developing formal methods for the cost forecasting of mechanical systems and their included components, it is important to identify the requirements of the engineer for cost modelling during the early phases of the design process. For the transformation from concept to an embodied solution, the purpose of an engineering cost model is really to describe the relative change in magnitude of costs within a component range and to provide a comparison of cost between component types. For the purpose of this work, a '*component type*' pertains to a particular engineering component such as a bearing or a gear, whilst a '*component range*' is considered to be either manufacturer specific or to distinguish between variants of a particular component type supplied by the same manufacturer. An example of component variants is in the case of single roller bearings, where flange configurations differ between ranges in order to provide for different combinations of axial and radial loading (SKF Limited, 1998).

In addition to component level costing, the designer must also consider system costs. Therefore, cost forecasting must provide sufficient accuracy upon which the costing of alternative design solutions and various configurations can be based. In order to achieve this, cost models for the full range of mechanical component types, from the various classes of engineering component need to be generated. In order to accommodate these various classes of engineering component; standard selected, standard designed and bespoke designed, four cost modelling approaches are discussed. Classes of engineering component and suitable cost modelling techniques are shown in figure 9.1. The overlap illustrated in figure 9.1 highlights the applicability of each technique to the three classes of component. This applicability is discussed in detail in the following sections.

9.2.1 Standard selected components

The ideal cost model for any third party component is a complete listing of current prices, although this may only be possible for standard selected components, such as bearings, actuators and motors. This class of component generally encompasses catalogued components, in which costing information may be explicitly defined for each component size. However, as detailed earlier in section 9.1, open access to costing information is rarely provided by suppliers. Therefore, for the majority of standard selected components, a specific cost model (SCM) described in detail in section 9.4, is an efficient method for the provision and generation of costing information. Here, a sample population is used to derive a statistical model for the cost of a particular component type. This cost model may then be used to generate relative cost for the full range of available component sizes.

9.2.2 Standard designed components

Standard designed elements are generally designed and sized by accepted numerical models, typically involving stress calculations, and are often restricted to standard size ranges, such as tooth modules or chain pitch. For this class of engineering component, a specific cost model (SCM) or a universal cost model (UCM) may be implemented, these are defined in sections 9.4 and 9.5 respectively.

In general, a specific cost model is derived from and applicable to a single component range from a manufacturer, whilst a universal cost model may be derived from one suppliers data, but may be applicable to another, or may describe more than one component range, i.e. a family of components. Such models are scalable to account for cost discrepancies between manufacturers or parametrically defined in order to describe various ranges within a manufacturers catalogue. During the early stages of the design process, capturing relative cost changes between elements in a range or between various component types is more important than capturing the exact cost, although acceptable levels of accuracy are desirable. It is considered by many practitioners and academics that an acceptable level of accuracy for a component cost model is one which captures between ±20 percent of the cost of the final artefact (Mileham *et al*, 1993).

9.2.3 Bespoke designed components

Bespoke elements are either manufactured in-house or out sourced. Often the high level of detailed design work required for such elements, demands that the designer possess a good appreciation of not only the geometric and performance details but also the processes and resources required for manufacture and production. It is for this class of component that configuration cost modelling (CCM) may be applied, discussed in section 9.6. In order to construct a configuration cost model the designer must possess knowledge of the product and production process, which includes materials, machining, and assembly operations (French, 1990). In the generation of a configuration cost model a parametric representation is constructed that relates the geometric and performance properties of the bespoke component to the specific costs for each of the three aspects of production previously mentioned. This provides for an indication of cost for various configurations of the particular component, enabling it to be included in a system modelling tool.

9.3 Developing cost relationships from component representations

For the purpose of this work, component based cost models relate cost to component attributes. These attributes may be selection attributes from a catalogue or features incorporated into bespoke elements. The generic procedure for developing cost relationships is shown in figure 9.2. Firstly, primary cost drivers must be identified. These are the attributes on which selection and specification of a particular component is based. Two or more primary cost drivers can be used if they are deemed independent and their contribution to the overall component cost can be clearly quantified. Otherwise, a single primary cost driver is used and secondary drivers applied to provide relative cost adjustment for changes in component attributes. For example, in the case of a bearing, the primary cost driver is chosen to be the dynamic load rating. Because this is generally the key criteria upon which suitable bearings are determined. The internal diameter is a secondary cost driver because the attributes are not deemed to be independent, that is their relative contribution to overall cost cannot be clearly identified. Therefore, this secondary cost driver (internal diameter) is used to adjust the relative cost for different dimensions.

In order to develop relationships for a cost model, a sample set of component specifications and cost information must be obtained. These should be dispersed over the full range with at least two or three readings for every change in magnitude of driver(s) variable or cost. An investigation into the relationship between component cost and the attributes of various mechanical components has identified five generic types of relationship that describes the majority of cost trends over the entire range of a component. These are illustrated in figure 9.3, they are not meant to be exhaustive but merely illustrate the common curves.

- Type 1 – Here the product cost is minimal over a subset of components, centrally located within the range. To either the side of this, the product cost increases, perhaps due to the reduced demand for components at the extremes of the product range.
- Type 2 – The typical 'j-shaped' cost curve that increases with product size. Here the increase in cost is probably due to both demand and increased materials and machining necessary as the product dimensions increase.
- Type 3 – The linear relationship is rare, although this is sometimes a good approximation to cost curves that contain many small fluctuations or oscillations in their form.
- Type 4 – This relationship exhibits a plateau in cost over a window mid range. This is likely to be demand driven, component batch sizes will be high and therefore relative production and overhead costs reduced.
- Type 5 – This is a common cost curve for catalogued components such as bearings. The form of the curve exhibits a number of steps. This is probably again due to demand and step changes in component dimensions, such as a change in the breadth of a bearing or the width of a gear.

All of these common relationships can be well defined using polynomial regression; the curves described are obtained by using polynomials up to and including a fifth order term for type 5. Although, where relationships exhibit a greater number of turning points than in type 5, a higher order polynomial may be necessary to capture the intricacies of the relationship. Ultimately, the statistical models used for cost relationships are dependent upon the domain and component. These examples provide for the typical relationships, the model developer may need to implement other common statistical methods, such as an exponential relationship or a logarithmic relationship (Foussier, 1998). For each of the three costing techniques described, this fundamental approach is applied in order to derive the cost relationship for and between various drivers.

Once the relationships for the cost drivers have been defined then the cost function can be constructed. The approach of this work is to generate system cost by combining the costs of all the constituent elements. This approach can be extended for individual components which possess more than one element or part, such as a chain drive, which is considered to encompass three main elements: two sprockets and a chain. The cost function may therefore contain a number of terms that represents each of these elements. This approach does not consider assembly costs, however, techniques have been developed by Boothroyd and Dewhurst Inc. that estimate assembly costs from the time taken to perform individual operations (Boothroyd & Dewhurst Inc, 2001).

9.4 Specific Cost Modelling (SCM)

A specific cost model quantifies the cost behaviour of a particular component *type* and *range*. For such models it is assumed that all components inside the range are manufactured utilising the same materials and processes. Consequently, only variables that describe the performance and geometry of the product are necessary in order to generate a cost model. The purpose of the cost model at the early stages of the design process is to provide costing information from key selection data. Therefore, the designer must select the appropriate driving variables for the cost model. These may be dimensions, weight, or performance capabilities such as loading and speeds.

9.4.1 A specific cost model for a chain drive

In this work the chain drive is considered to encompass three main elements: a driver sprocket, a driven sprocket and a chain. The cost forecasting approach combines the cost of each element to provide an indication of the cost for the chain assembly when procured. Three cost drivers are identified: the number of teeth, the pitch and the chain length. Three cost curves are used; number of teeth versus cost, pitch versus relative cost for a sprocket, and pitch versus cost per metre of chain. These characteristics are shown in figure 9.4.

Polynomial regression is used to fit curves to the sample data. The terms generated by the curve fitting are

$$Cost_{np} = 39.51 + (-3.44 \times N_{teeth}) + (0.16 \times N_{teeth}^2) + (-2.52 \times 10^{-3} \times N_{teeth}^3) + (1.30 \times 10^{-5} \times N_{teeth}^4)$$

where N_{teeth} is the number of teeth and $Cost_{np}$ is the cost of the sprocket for a nominal pitch. And

$$Cost_{relative} = -0.31 + (0.18 \times Pitch) + (-1.66 \times 10^{-3} \times Pitch^2)$$

where $Cost_{relative}$ is the relative cost of a sprocket against a nominal pitch and *Pitch* is the sprocket pitch. And

$$Cost_{metre} = 33.08 + (-3.12 \times Pitch) + (10.05 \times 10^{-3} \times Pitch^2)$$

where $Cost_{metre}$ is the cost per metre of chain for the given pitch.

It follows that,

$$Cost_{sprocket} = Cost_{np} \times Cost_{relative}$$

And

$$Cost_{chain} = Cost_{metre} \times Length_{chain}$$

where $Length_{chain}$ is the length of the chain in metres.

Therefore, the overall cost is equal to

$$Cost_{overall} = Cost_{sprocket1} + Cost_{sprocket2} + Cost_{chain}$$

9.4.2 Results

The cost model generated in section 9.4.1 is based on sample costing data provided by Fenner (1999). The model is validated through comparison of predicted cost against actual cost for a number of chain assemblies, illustrated in figure 9.5. In order to generate meaningful results, it is important that the sample data used for developing a cost model and the test data used to validate the model are separate and distinct data sets. The results of the test cases highlight a strong correlation between predicted costs and actual costs, with a mean percentage error for predicted costs of 11 percent. This value is considered to be acceptable for representing the component pricing structure and is therefore suitable for decision making at the early stages of product design.

9.5 Universal Cost Modelling (UCM)

A universal cost model is used in situations where it is necessary to represent the costing structure of either a generic type of mechanical component or a family of like products. For representing a particular component type a single fundamental cost model may be generated, but scaled to account for changes in magnitude of cost from different manufacturers. In the case of product families, a single cost model maybe used and its coefficients parametrically defined in order to accommodate the various ranges. For example, in the case of bearings, various ranges of single roller bearings are available with different flange configurations for various combinations of radial and axial loading.

Universal cost models utilise the same fundamental model building principles as specific cost models. These include the identification of the driving variables and the application of appropriate mathematical models. Once an appropriate model has been determined its coefficients are parametrically defined. For a number of sample ranges the model must be applied and the coefficients determined. The relationship between corresponding coefficients for the various ranges may then be investigated. This relationship may be adequately described through linear interpolation or may require complex non-linear relationships. The advantage of this approach is that a single cost model can be used to describe multiple ranges, or component types from different manufacturers, by simply altering the model parameters.

9.5.1 A universal cost model for a standard gear range

To illustrate the general points made above, this section develops a universal cost model for different ranges of gears from a particular manufacturer. These gear ranges are available for various tooth modules and for the manufacturer considered, the primary cost driver for the gear ranges is the number of teeth (Shigley, 1983). Another attribute which significantly affects cost, is the gear face width, which for the example considered is predetermined and related to increases in tooth module. Therefore, the driving variable for the parameterisation of the coefficients for the cost model is taken to be the tooth module.

In order to develop a cost model, various curve fitting techniques were applied to the sample cost data for different gear ranges. Of the fits investigated, the best overall fit for the sample gear ranges was identified as a 5th order polynomial, shown in part (a) of figure 9.6. The best fit is determined using the statistical method termed the product moment correlation coefficient, a positive value tending to 1 is deemed a good fit or strong correlation (Spiegel, 1961). Application of a parameterised 5th order polynomial to the sample gear ranges generates the various coefficients. The relationships between corresponding coefficients from each sample range have been investigated and various curve fitting techniques used to describe the variation in coefficients for different ranges, examples of which are shown in parts (b) and (c) of figure 9.6.

It follows that the terms for the parameterised cost model are

$$Cost_{gear} = A + B_1 \times N_{teeth} + B_2 \times N_{teeth}^2 + B_3 \times N_{teeth}^3 + B_4 \times N_{teeth}^4 + B_5 \times N_{teeth}^5$$

where $Cost_{gear}$ is the gear cost, A and B_1 to B_5 are parametric coefficients defined by

$$A = 2.727 + 4.023 \times Module_{gear} 2.73{+}4.02$$

$$B_1 = 0.254 + (-0.388 \times Module_{gear}) + (0.026 \times Module_{gear}^2)$$

$$B_2 = -0.014 + (0.031 \times Module_{gear}) + (-4.011 \times 10^{-3} \times Module_{gear}^2)$$

$$B_3 = 0.244 \times 10^{-3} + (-0.67 \times 10^{-3} \times Module_{gear}) + (0.199 \times 10^{-3} \times Module_{gear}^2)$$

$$B_4 = 3.143 \times 10^{-6} + (9.224 \times 10^{-6} \times Module_{gear}) + (-3.82 \times 10^{-6} \times Module_{gear}^2)$$

$$B_5 = 15.88 \times 10^{-9} + (-50.29 \times 10^{-9} \times Module_{gear}) + (26.77 \times 10^{-9} \times Module_{gear}^2)$$

9.5.2 Results

The cost model generated in section 9.5.1 is based on sample costing data provided by HPC Gears (1997). The model is validated through comparison of predicted cost against actual cost for a sample of gears from different ranges, illustrated in figure 9.8. The results of the test cases highlight a strong correlation between predicted costs and actual costs, with a mean percentage error for predicted costs of 10 percent. This value is considered to be acceptable for capturing the relative variation in component cost of the gear ranges and for reliably describing the cost of the component.

9.6 Configuration Cost Modelling (CCM)

Configuration cost modelling aims to capture the costs incurred through changes in the parameters or attributes of a bespoke designed component. In the case of bespoke components, it is generally accepted that the type of features and often the number of features of the component will remain constant as the component parameters change, although the extent of these features may vary. Consequently, the approach of configuration cost modelling is to represent the costs incurred through changes in the extents of these features brought about by changes in the primary attributes of the component. Theses primary attributes are the key attributes upon which a component is selected or specified (Culley & Allen, 1999). In the case of a shaft these may include the number of nodes or sections, their respective diameters and lengths. In order to represent the cost associated with changes in these attributes, it is necessary to quantify the relative changes in materials, machining and any assembly operations. Therefore, the cost of a bespoke designed component can be generated by combining the cost of materials, machining and assembly. Thus, the cost may be expressed in the form

$$Cost_{component} = Cost_{assembly} + Cost_{machining} + Cost_{materials} \text{ (French, 1990)}$$

$$Cost_{component} = G_a(F_a) + G_m(F_m) + G_{ma}(F_{ma})$$

where G_a, G_m and G_{ma} are the specific costs of assembly, materials and machining respectively, and F_a, F_m, F_{ma} are functions which describe the levels of materials, machining and assembly, and are comprised from primary component attribute descriptors. These are either geometric or performance related. Utilising this approach enables the designer to construct a model which predicts the cost of production for a particular component. This may also be used to provide costing for a component that is procured from a third party, providing that the model builder possesses adequate knowledge about the production process for the component and specific costing.

9.6.1 A configuration cost model for a shaft

In this work, the primary drivers for the cost model of a shaft are the number of sections, the relative dimensions of each section and the material. The three cost terms for the model are described below and their associated specific costs are derived. The total cost for production of the shaft is given as

$$Cost_{shaft} = Cost_{assembly} + Cost_{machining} + Cost_{materials}$$

For the purpose of this work, the assembly costs for a shaft are assumed to be similar for different configurations and therefore do not need to be considered. However, there will be significant time allocated to set up for machining operations on the shaft, it thus follows that

$$Cost_{shaft} = G_m(F_m) + G_{ma}(F_{ma})$$

where $Cost_{shaft}$ is the shaft cost, the coefficients G are the specific costs and F are the terms associated with the level of resources for each aspect of production for the shaft.

The function F_m and specific cost of the material G_m are derived from

$$F_m = d \times \sum_{i=1}^{n} \frac{\pi \times Diameter_i^2}{4} Length_i$$

where $Diameter_i$ and $Length_i$ are the diameter and length of each shaft section or node and d is the density of the material. And

$$G_m = 4.21 + (-12.49 \times 10^{-3} \times \sigma_{shaft}) + (15.3 \times 10^{-6} \times \sigma_{shaft}^2)$$

where σ_{shaft} is the yield strength of the shaft material.

The function F_{ma} and specific cost for labour, G_{ma} are derived from

$$F_{ma} = \left(d \times \sum_{i=1}^{n} \frac{\pi \times (Diameter_{\max}^2 - Diameter_i^2)}{4} Length_i \times C_1 \right) + \left(d \times \sum_{i=1}^{n} Length_i \times C_2 \right) + setup$$

where $Diameter_i$ and $Length_i$ are the diameter and length of each shaft section or node, $Diameter_{max}$ is the maximum shaft diameter and *setup* is the time taken for machine and tool setup. The function provides for roughing cuts and a finishing cut. The coefficients C_1 and C_2 are the material rate of removal and feedrate respectively.

$$C_1 = 1 \Bigg/ \left(\frac{\pi \times (Diameter_{\max} + Diameter_i) \times d \times f \times N}{2} \right)$$

$$C_2 = \frac{1}{f}$$

where f is the feed rate, d is the depth of cut and N the speed.

G_{ma} = Cost per hour for an operator

The model developed in this work is based on production rates and labour costs supplied by industry. These represent arbitrary values, which can easily be changed to reflect local circumstances.

9.7 Integrated cost modelling for systems

The purpose of integrated cost modelling for mechanical systems is to enable the generation of an indication of cost for different design configurations. Where these configurations might be different arrangements of components, different component types, sizes and various combinations. The ability to represent and manipulate design configurations is provided by a component based system modelling environment described in chapter 8. This modelling environment integrates electronic representations for various mechanical components and enables the physical and performance dependencies between elements of the system to be considered in some detail. The ability to represent these dependencies ensures that a system of physically compatible (in terms of connectivity) and mutually compliant (in terms of their performance capabilities) is determined. This approach provides the performance and geometric attributes necessary for the application of the cost models.

The consideration of the system and its included elements as a whole is essential for the strategic design and optimisation of machine systems. For example, consider a shaft and a bearing, shown in figure 9.8. Part (a) shows the shaft cost against diameter and part (b) shows the bearing cost against internal diameter. More often than not, the designer aims to optimise the considered system against cost. For the example considered it may seem appropriate to minimise the shaft diameter to reduce material and the size of bearing required, and hence reduce costs. However, the bearing cost actually rises as the bearing becomes miniaturised. This dependency is shown in part (c), where a cost optimised solution is not to miniaturise the shaft diameter. Figure 9.9 illustrates cost, mass and leading dimensions for an arbitrary system for a number of different component configurations. Whilst each solution is feasible, it is clear that to minimise mass or leading dimensions of the assembly would incur significant costs. It is therefore essential that costs are considered during the strategic design and optimisation of systems.

The integrated modelling environment enables the construction of a schematic representation of a concept, shown in figure 9.10. Governing models for each component may then be selected, the system requirements are specified by the user and the system model resolved. This resolution process generates and manipulates selection data for the execution of component based representations. Once resolved, a complete set of real and compatible elements is generated and selection parameters for individual elements are produced. These selection parameters provide the values necessary for the execution of cost models. The cost modelling techniques described in this chapter have been implemented for a range of mechanical elements within the modelling environment. These include; specific cost models for a chain drive, written in BASIC; a family cost model for gears in a spreadsheet environment and configuration cost models for shafts and keyways, coded in C.

The costing modules are executed each time a component model is interrogated. In this manner, system cost can be evaluated for a variety of configurations. An example configuration is shown in figure 9.10. The introduction of cost modelling techniques into the system modelling approach provides for the strategic design of systems against cost, spatial occupancy and performance criteria. All of which are important considerations for the design of any mechanical system. The application of the modelling tool to a case study is shown in figure 9.10. The predicted cost of the assembly is compared to the actual cost, obtained from suppliers, shown in figure 9.11. The results show that for individual components the predicted cost is within 80 percent of their actual cost, whilst the predicted cost for the overall system is 96 percent of the actual cost of the system. The accuracy shown in this example is largely due the large number of cost models that are based on real production data and as a result no errors are included.

9.8 Concluding remarks

In today's aggressive global market place, companies are under increasing pressure to produce high quality low-cost design solutions. In the case of mechanical systems, the designer has the greatest influence over the cost of the final designed artefact during the early design phases.

Consequently, the designer requires supportive methods and techniques to enable comparative studies of design alternatives and to produce reliable indications of performance and cost as soon as is practicable. Because electronic representations tend to restrict access to costing data, true cost, can increasingly only be obtained by contacting the supplier directly. To address this issue, and enable the consideration of cost within the modelling approach, a range of cost modelling techniques for the various classes of engineering components; standard selected, standard designed and bespoke designed have been developed. These techniques are created in order to provide cost forecasts from the selection attributes of a particular component. The models aim to represent or capture the cost structure within a component range in order that various sizes can be considered by the designer, and comparisons made between various component types and combinations. For this purpose, cost models are considered to be suitable techniques for supporting the decision process if they represent the cost of a component to within 20 percent.

The introduction of the cost modelling techniques into the integrated modelling environment enables an indication of the cost of systems to be generated. With the inclusion of cost information the environment provides for the consideration of spatial occupancy, performance, mass and costing. The ability to consider all of these aspects assists the designer in selecting a suitable design configuration and embodying the configuration with an optimum set of components. Both of which are essential elements for the successful design of a product. Furthermore, the availability of cost data provides a platform for the application of optimisation techniques, which are discussed in chapter 10.

<table>
<tr><th>Class of Component</th><th colspan="2">Type of Cost Model</th></tr>
<tr><td rowspan="2">Standard selected
selected from a range of component sizes held by third party suppliers.</td><td colspan="2">Listed
complete listing of current prices for a range of components</td></tr>
<tr><td rowspan="2">Universal
represent the cost structure of a type of mechanical component or a family of like products</td><td rowspan="2">Specific
represents the cost behaviour of a particular component type and range</td></tr>
<tr><td>Standard designed
fully specified through practiced or established design procedures.</td></tr>
<tr><td>Bespoke designed
one-off components tailored to meet specific requirements for the application considered</td><td colspan="2">Configuration
Captures the relative costs incurred through changes in the attributes of a particular component</td></tr>
</table>

Figure 9.1 – Classes of engineering components and cost modelling techniques

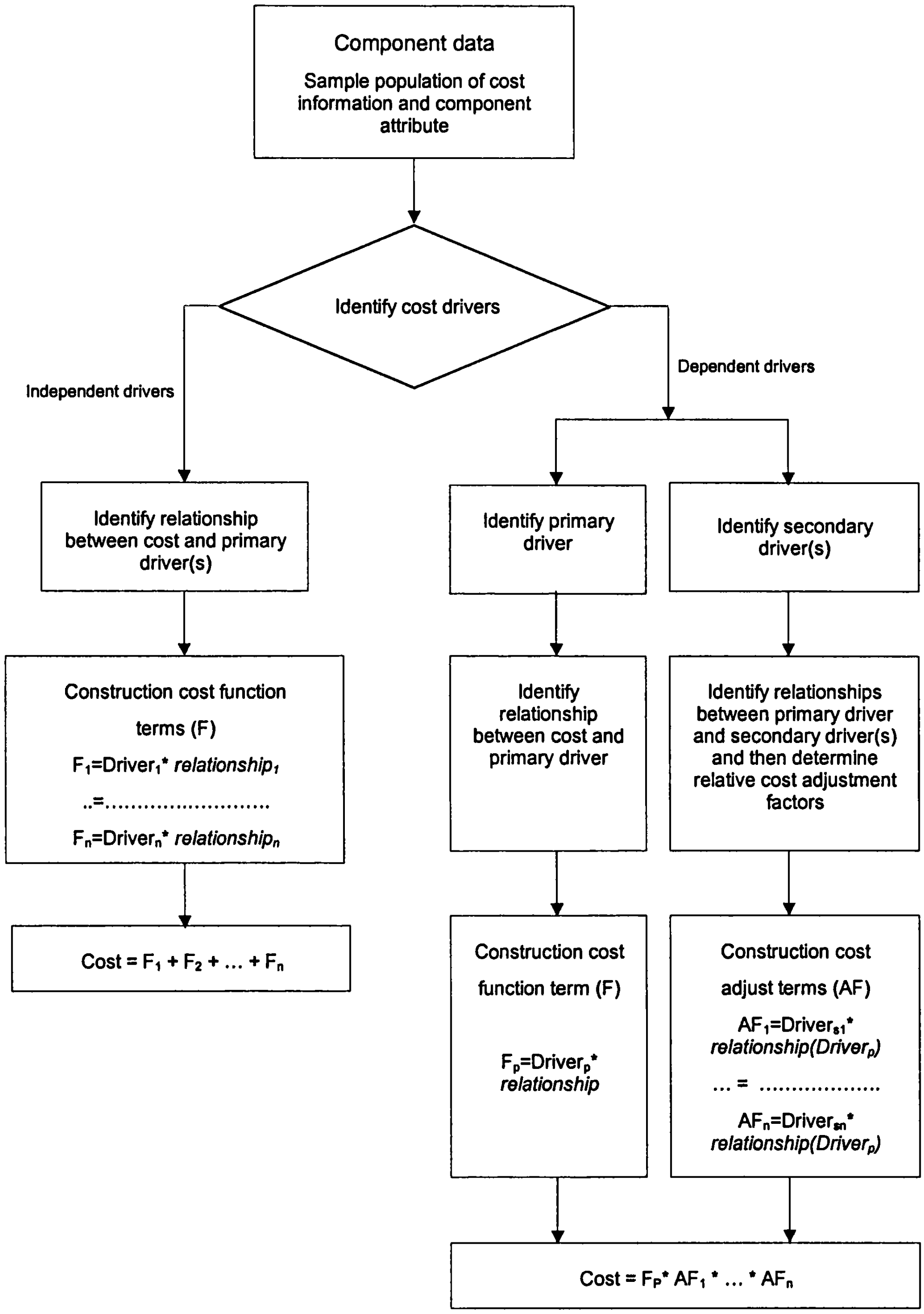


Figure 9.2 – Generic process for the development of cost relationships

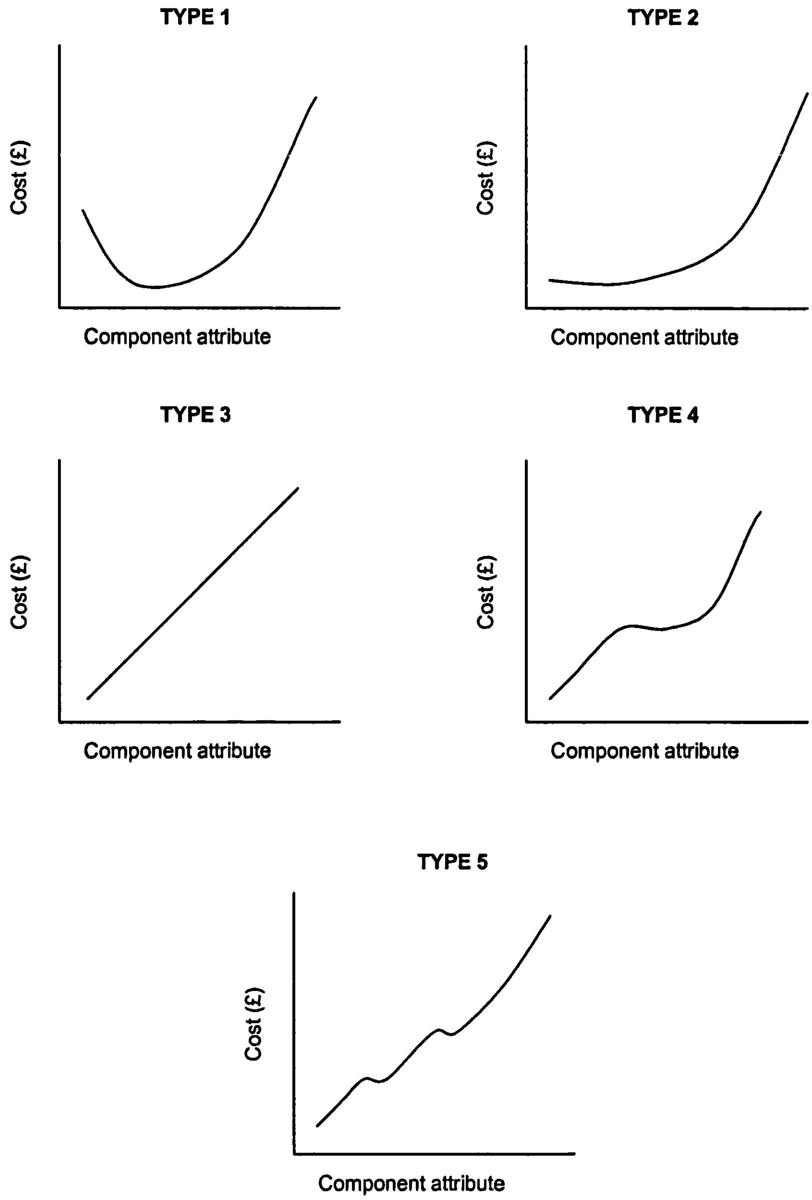
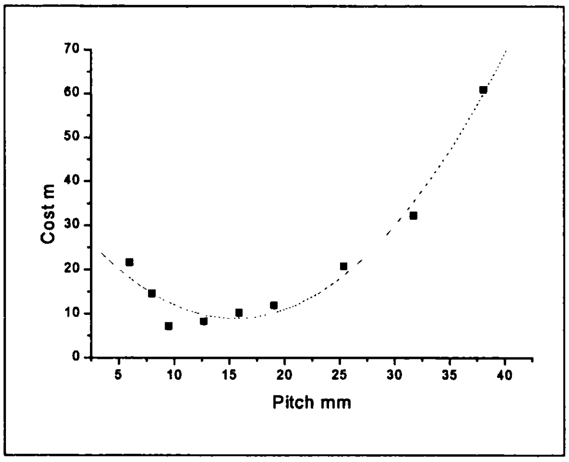


Figure 9.3 – Typical relationships for cost and various component attributes

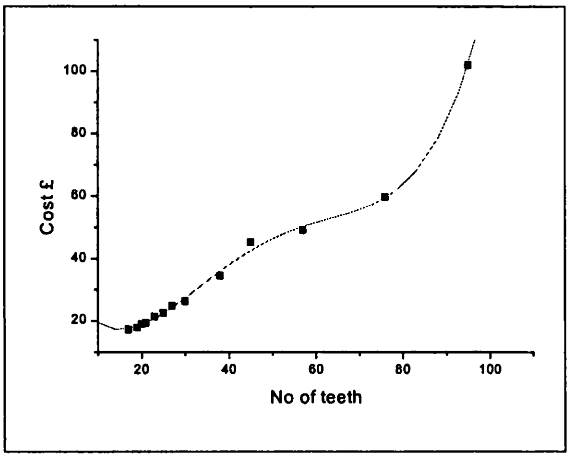


Regression curve

$Y = A + B1*X + B2*X^2$

Parameter	Value
A	33.07982
B1	-3.12025
B2	0.10047

Part (a) Cost per metre of chain for various pitches

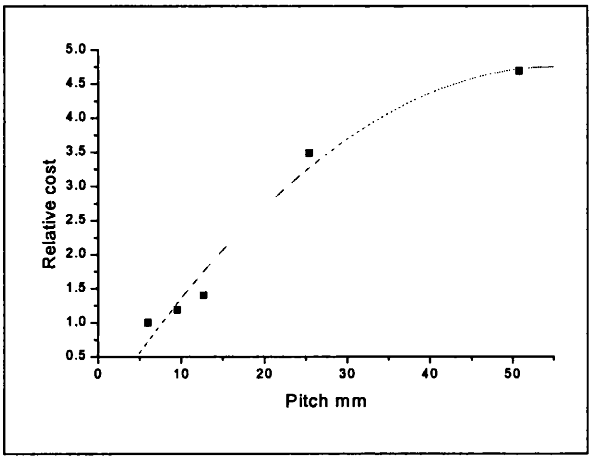


Regression curve

$Y = A + B1*X + B2*X^2 + B3*X^3 + B4*X^4$

Parameter	Value
A	39.50978
B1	-3.43874
B2	0.16494
B3	-0.00252
B4	1.30E-05

Part (b) Cost of sprocket for number of sprocket teeth (for nominal pitch)



Regression curve

$Y = A + B1*X + B2*X^2$

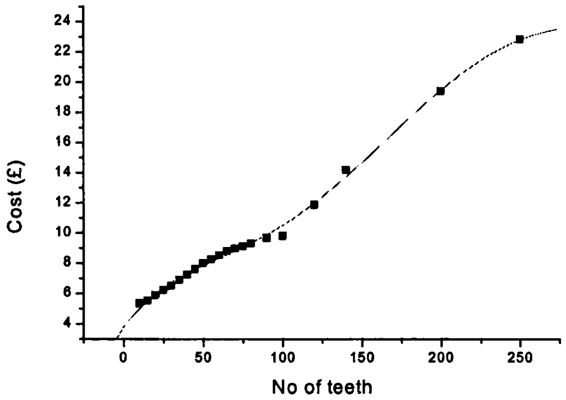
Parameter	Value
A	-0.30967
B1	0.18317
B2	-0.00166

Part (c) Relative costing for sprocket pitch

Figure 9.4 – Polynomial cost curves for a specific chain drive cost model

Assembly specification				Actual cost	Predicted cost	Percentage error
Pitch	Sprocket 1	Sprocket2	Chain length			
9.5	17	30	952	50.28	59.30	17.94
12.7	20	57	1955	108.85	126.43	16.15
15.875	13	38	1618	101.27	90.29	10.84
19	16	57	2780	194.91	200.35	2.79
25.4	15	28	3276	222.31	202.22	9.04
					Mean Error	11.35

Figure 9.5 – Comparison of predicted and actual component costing

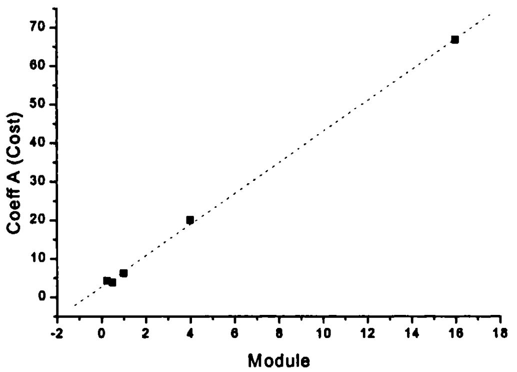


Regression curve

$Y = A + B1*X + B2*X^2 + B3*X^3 + B4*X^4 + B5*X^5$

Parameter	Value
A	3.80184
B1	0.14283
B2	-0.00197
B3	1.73186E-5
B4	-5.89133E-8
B5	6.74508E11

Part (a) – 5th order polynomial cost curve for a tooth module of 0.5

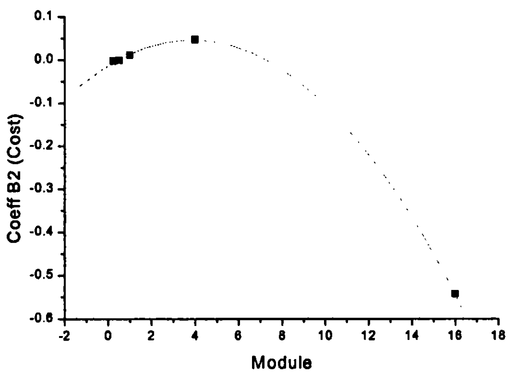


Regression curve

$Y = A + B1*X$

Parameter	Value
A	2.72676
B1	4.02281

Part (b) – Relationship between coefficients A from part (a) for various gear modules



Regression curve

$Y = A + B1*X + B2*X^2$

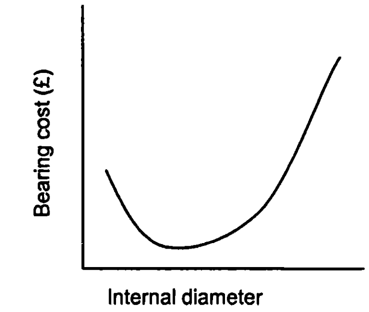
Parameter	Value
A	-0.01333
B1	0.03087
B2	-0.004

Part (c) – Relationship between coefficient B2 from part (a) for various gear modules

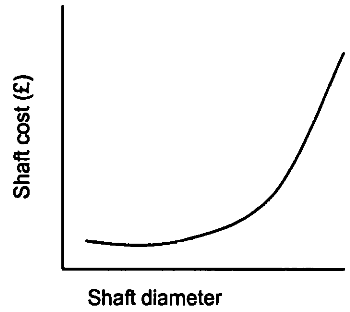
Figure 9.6 – Polynomial cost curves for a universal cost model of a gear pair

Gear specification		Actual cost	Predicted cost	Error	
No of teeth	Module			Value	Percentage
20	2	12.15	11.66	-0.49	4.03
30	2	15.9	17.55	1.65	10.38
50	2	27.09	35.57	8.48	31.30
20	3	18.7	17.55	-1.15	6.15
40	3	36.97	43.7	6.73	18.20
70	3	102.99	131.32	28.33	27.51
20	8	82.34	79.4	-2.94	3.57
30	8	154.21	139.65	-14.56	9.44
45	8	237	218.68	-18.32	7.73
				Mean error	10.06

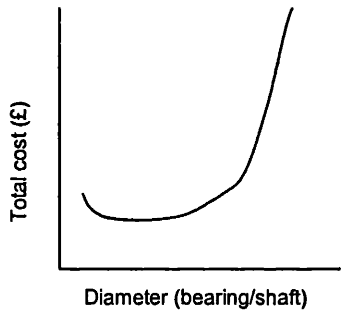
Figure 9.7 – Comparison of predicted and actual component costing



Part (a) bearing cost against internal diameter



Part (b) shaft diameter against shaft cost



Part (c) shows combined cost of shaft and bearing

Figure 9.8 – Example cost curves for a bearing, a shaft and a combined bearing shaft assembly

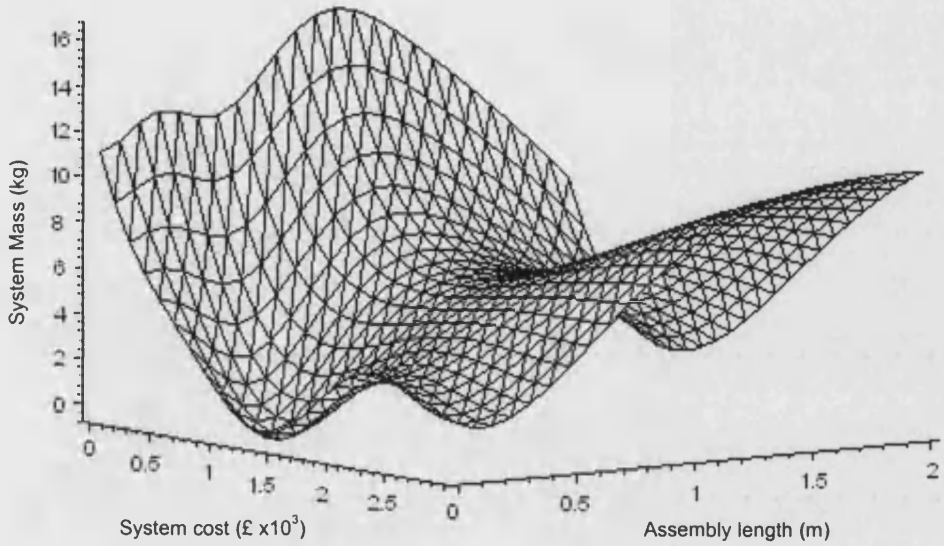


Figure 9.9 – System attributes for a range of system configurations

Part (a)

Part (b)

Part (c)

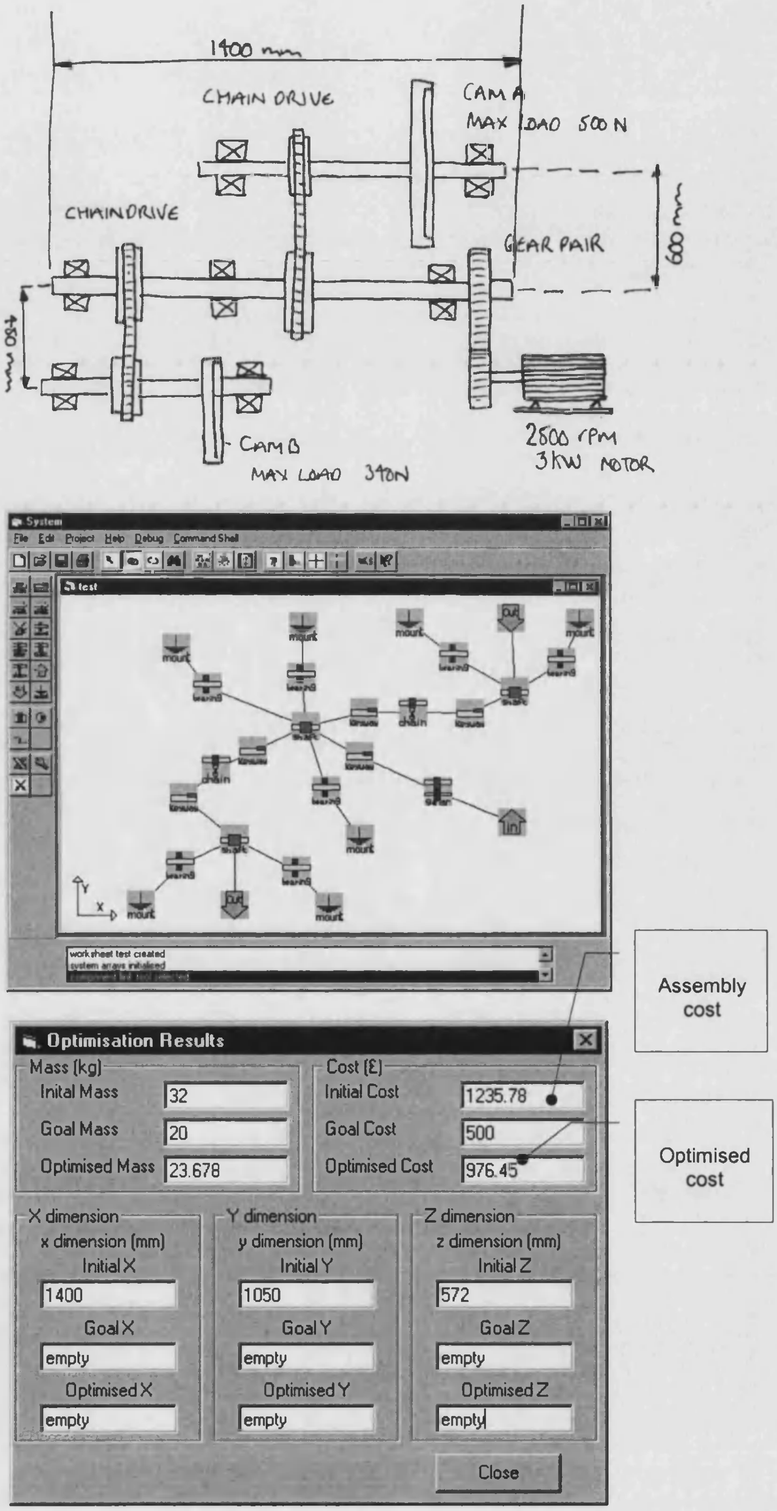


Figure 9.10 – The embodiment of a drive-system for an over wrapper with consideration of cost

Component	Cost model	Predicted cost	Actual cost	Percentage error
bearing 1	SCM	38.43	40	4.09
bearing 2	SCM	68.24	67	1.82
bearing 3	SCM	29.5	34	15.25
keyway 1	CCM	4.71	4.71	Note 1
keyway 2	CCM	4.83	4.83	Note 1
keyway 3	CCM	9.79	9.79	Note 1
shaft 1	CCM	40.86	40.86	Note 1
gear	UCM	104.01	126.76	21.87
chain drive 1	SCM	79.4	87.06	9.65
chain drive 2	SCM	139.5	117.78	15.57
shaft 2	CCM	30.33	30.33	Note 1
bearing 4	SCM	27.81	33	18.66
bearing 5	SCM	30.42	35	15.06
keyway 4	CCM	5.85	5.85	Note 1
keyway 5	CCM	19.72	19.72	Note 1
shaft 3	CCM	29.98	29.98	Note 1
bearing 6	SCM	23.21	27	16.33
bearing 7	SCM	25.88	31	19.78
keyway 6	CCM	23.68	23.68	Note 1
keyway 7	CCM	4.93	4.93	Note 1
cam 1	Procured	140	140	Note 2
cam 2	Procured	160	160	Note 2
	Total cost	1041.08	1073.28	3.09

Note 1 – Costs provided by machinery manufacturer for one off cams CNC milled

Note 2 – Costs derived from actual cost data

Figure 9.11 – Cost comparison of a drive-system for an over wrapper

Chapter 10

Optimisation issues in an integrated modelling environment

It has be shown in previous chapters how engineering systems can be modelled and then populated with mechanical components by combining and manipulating various electronic representations. This modelling approach enables the consideration of performance, spatial occupancy, mass and through the incorporation of cost modelling techniques an indication of relative cost can also be generated. The determination of these attributes for a system model provides an extensive and important range of attributes which can be used to optimise the system. In fact, the application of optimisation (search) techniques for systems comprising standard components is a particularly important area because of the combinatorial problem and the complex solution space (Ward & Seering, 1989; Theobald, 1995).

This chapter discusses the issues associated with the difficult problem of optimising mechanical systems based on standard designed and standard selected mechanical components. These issues will be used to identify the requirements for optimisation and to develop a strategy for system optimisation. This problem is made more difficult by the fact that various third party electronic representations are used to determine an overall solution, and therefore data necessary for optimisation is only implicitly represented in the modelling environment.

The work described does not attempt to develop an optimisation algorithm *per se* rather it reviews and identifies a suitable optimisation algorithm for incorporation into the integrated modelling approach. In a similar manner to the development of the integrated modelling approach, the strategy for optimisation aims to provide an interface between the modeller and a third party optimisation tool. This would enable the incorporating of various techniques and optimisation algorithms in the modelling approach.

10.1 Design optimisation

Design is essentially an open-ended problem, the selection of the 'best' or 'optimum' solution has always been and continues to be a concern of the designer. Over the last decade much work has been undertaken which deals with the application of mathematical optimisation techniques in engineering design. Furthermore, commercial design packages with optimisation capabilities are now widely available, such as INCA (INBIS Group Plc, 2001), iSight (Engineous Software Inc, 2001) and MathWorks (The MathsWorks Inc, 2001), which all provide optimisation tools for complex systems and processes.

In engineering design, optimisation is often used to develop or identify the best performing solution where this best performance is represented (measured) by an objective function. The formulation of this objective function is dependent on the design problem and the individual constraints imposed on the particular problem. There are many and varied types of optimisation objectives or goals in engineering design, ranging from shape or form optimisation, structural, reliability and cost (Adeli, 1994). The formulation of the objective function is critical to the success of any optimisation approach, and is dependent on the availability of parameters within the modelling and optimisation environment. In fact, a limitation of many approaches is that they consider only a single variable, such as cost, when in actual fact the designer will rarely optimise on a single factor such as cost. More often, there will be a number of considerations which represent a combination of performance, geometry and cost for the considered system, termed multi-objective optimisation. In these cases, the optimisation algorithm endeavours to determine an optimum trade-off between the two or more components or terms of the objective function. Section 9.8 highlights the importance of considering multiple system attributes during optimisation. For the purpose of this work, and in particular the optimisation of systems of standard components three objectives for optimisation are identified: cost, mass and spatial occupancy.

In the area of system optimisation with standard components some work has been undertaken by Ward and Seering (1989) and by Theobald (1995). The optimisation of systems containing standard components is complicated by the non-uniform discretisation of component sizes across a particular range. Furthermore, during optimisation the limits or bounds of component ranges must also be considered. Theobald acknowledges the existence of discrete regions within the overall solution space. However, the work focused on developing modelling techniques to represent standard selected elements. The majority of these techniques generate continuous parametric models that at best approximate the discrete nature of the engineering components.

Work by Ward and Seering considers only catalogued components and uses a process of constraint propagation to eliminate sets within each catalogue. This elimination is facilitated by the hierarchical structure of the catalogues incorporated within the design environment. This constraint propagation, produces a variety of alternative component combinations. Possible alternatives are evaluated by repeatedly splitting catalogues in a sequential manner. This generates a binary best-fit search tree. The compiler always splits the leaf tree offering the lowest possible cost and as a consequence a cost optimised solution is only ever developed.

Both these approaches involve the creation of either an abstracted or restructured model for individual components in order to enable or improve optimisation. This inability to utilise existing component models may result in a solution space, that at best, can only represent feasible or approximate component sizes, which for solutions that may be sensitive to changes in component specification and in particular geometry is all but useless. Whilst this work does not seek to develop an optimisation algorithm it does aim to consider real components from third party electronic representations. In order to achieve this, a suitable optimisation algorithm is identified and altered for the particular application. This alteration is necessary for multi-objective optimisation of a constrained problem containing discrete elements within the overall solution space. This optimisation algorithm is encapsulated in a third party software application, and requires that a strategy is developed to allow for interfacing the optimisation software with the modelling environment.

10.2 Optimisation methods

Optimisation involves either minimising or maximising a function f, in this case the minimisation of the function. There are two types of problem, constrained and unconstrained. Constrained problems restrict the solution space to variables that are within a certain range or that satisfy a governing function or functions. Unconstrained problems permit any value of parameter between $-\infty$ and ∞ and do not impose implicit restrictions.

Optimisation algorithms operate by repeatedly altering specific variables, known as design variables, and evaluating the effect on the objective function. The algorithms generally evaluate preliminary or exploratory changes in parameters and identify those which yield a positive effect on the objective function. These successful changes are then pursued. The major difference between many optimisation routines is the manner or method in which the exploratory changes are made and how successful solutions are used to determine a new search region. The move to the new search region is usually termed a pattern move. In general, two distinct classes of optimisation methods exist; direct and indirect search methods (Walsh, 1975).

10.2.1 Indirect search methods

This class of algorithm endeavours to locate an optimum solution by examining the properties of the function and often its derivative around the point of interest. In order to implement this class of algorithm the objective function must be defined explicitly. However, for the systems modelling approach developed in this work, the various components that constitute the objective function are represented implicitly within the third party models. Consequently, the function cannot be explicitly evaluated and this class of algorithm is inappropriate for the given problem.

10.2.2 Direct search methods

Direct search methods do not require derivative values of the objective function and hence do not require the explicit definition of the function. These algorithms generally begin at an initial solution and either logically (deterministic) or randomly (stochastic) step through the solution space in order to determine an optimum solution. Common algorithms, their various applications, advantages and drawbacks have been discussed by many authors over the last few decades such as Walsh (1975) and Hajela (1999). However, for many of the techniques, such as Genetic Algorithms, the algorithm must be tailored for each application, demanding prior knowledge of the solution space. For the problem of mechanical systems comprising standard components, the solution space cannot be easily determined and may in some instances comprise only continuous elements, only discrete elements or varying combinations of each. Because of this, the general-purpose method of Hooke and Jeeves (Walsh, 1975), which is well suited to constrained problems and uncertain solution spaces, is used for optimising this problem.

10.2.3 Hooke and Jeeves

The Hooke and Jeeves method dates from 1961. The approach attempts to determine an optimum solution by using a simple strategy to determine the best search directions from an initial base point. The strategy comprises two elements, exploratory moves and pattern moves. Exploratory moves aim to acquire information about *f(x)* in the region of the current base point. A pattern move is effected from the information obtained during the sequence of exploratory moves.

Consider a function *f(x)*. The search commences from an initial base point b_1 and step lengths h_j are selected for the respective variables x_j and e_j the unit coordinate vector. The step length is typically determined on the basis that the value of $|F(b + h_j e_j) - F(b_1)|$ is approximately zero due to a change of one step length in each variable in turn. The method proceeds by a sequence of exploratory and pattern moves.

Exploratory moves acquire information about *f(x)*in the region of the current base point. The procedure for an exploratory move about base point b_1 is to evaluate $F(b_1 + h_1 e_1)$. If the move from b_1 to $b_1+h_1e_1$ is a success replace base point b_1 by $b_1+h_1e_1$. If it is a failure then evaluate $F(b_1 - h_1 e_1)$. This is repeated for each variable in turn, considering moves $\pm h_j e_j$ for a base point which results from the previous variable. So for the case of x_2 the base point may be $b_1+h_1e_1$. Once this has been completed for each variable, then either a new base point b_2 is selected and a pattern move undertaken from this point, or step lengths are changed and the process of exploratory moves repeated.

A pattern move attempts to speed up the search by using information already acquired about f(x). If p_1, p_2 ... p_n denote the points reached by successive pattern moves for base points b_2, b_3 ... b_n. The pattern move for b_2 is as follows; $p_1 = 2b_2 - b_1$. At this new point a series of exploratory moves are executed about p_1. If the value of the function during these exploratory moves is less than $f(b_2)$ then a new base point b_3 is reached and a pattern move about b_3 is executed. This process is repeated until either a predefined number of iterations have been undertaken or a threshold value for the function has been reached.

Many optimisation methods follow a similar overall process. The main difference is the manner in which the pattern move is effected. Some methods utilise a quadratic function which is fitted through a number of previous base points in order to determine the next move. Methods such as Powells actually transform the local search coordinates to align with the direction of the previous pattern move. This enables exploratory moves to be made in the direction of the current search path and perpendicular to it (in the case of two variables).

10.2.4 Optimisation software

For the purpose of this work, a constraint modelling system called RASOR (Rules And Systems Of Rules) (Medland *et al*, 1996) is incorporated into the approach. The environment utilises a direct search technique to find a solution that satisfies the imposed constraints (Medland, 1990). These constraints can operate on geometry and mathematical relationships and form the objective function. This function consists of constraint rules that may contain any equality, the value of which is minimised when the set of constraints is solved. This enables either specific values or thresholds to be set for functions, or for the value of the function to be minimised. The environment currently provides for two optimisation algorithms; Hooke and Jeeves and Powells method. The application requires that a macro is constructed, depicted in figure 10.1. This macro declares the parameters and the constraint rules and sets up the optimisation problem. In addition to this, the RASOR system is selected because the environment incorporates full DDE capabilities, which enables interprocess communication (Bowler *et al*, 1999). In this manner, the optimiser can be interfaced with the integrated modelling environment using a similar approach as with the electronic representations, discussed in chapter 7.

10.3 An optimisation strategy

In the development of a strategy for system optimisation four key areas are addressed. These relate to the problem formulation, dealing with discrete elements, model resolution during optimisation and the development of a software architecture for optimisation.

10.3.1 Problem formulation

During problem formulation the objective function and the design variables for optimisation are set. As discussed in section 10.1, the formulation of the objective function is critical and must accurately reflect the design requirements. For the purpose of this work, three terms for inclusion in the objective function are identified; mass, cost and spatial occupancy. These are the primary quantitative system attributes that may be calculated from the formal data that specifies a particular mechanical component. For many components, costing information has to be modelled and techniques for this are discussed in chapter 9. Spatial occupancy is determined by evaluating the leading dimensions of the system. Data describing the centroid and spatial envelope of each component are produced during system resolution, discussed in chapter 4. This data can be combined to generate the overall dimensions for the system. For the purpose of this work, multiple objectives can be set and the goal can be explicitly defined to either minimise each term of the objective function or achieve desired values for certain terms of the objective function. For this latter case, the goal is to minimise the error, which is the difference between the actual value for the term and the desired value. The interface for the construction of the objective function is depicted in figure 10.2.

In addition to constructing the objective function the designer must identify the design variables. These are the parameters which the optimser will use to search for an improved solution. The specification of these parameters is also critical to the success of the optimisation episode. Although much work has been undertaken in the area of identifying design variables, through techniques such as sensitivity analysis (Adleli, 1994). There are still no formalised rules for the identification of these design variables. For the majority of situations, the designer must possess an understanding of the problem and an appreciation of the mathematical representations that govern them. Although parametric representations are necessary for optimisation, a drawback of many parameterised models is that models are constructed so that they can only be driven by a number of specific variables. Thus, in these instances only these specific variables are useful for optimisation and only a particular optimised solution is generated for the configured model (Papalambros & Wilde, 1988).

For the purpose of this work, the greatest effect on the system and components can be obtained by altering the parameters of the core components. This is because there is the highest level of dependency (coupling) between core elements and other components in the system because of the large number of connections.

During the embodiment of systems, one of the most important considerations is does the solution exist, i.e. can the components be procured. For the purpose of this work, a solution is only considered feasible if a system of real[1] components is determined. The identification of a set of 'real' components is made more difficult by the predominance of standard selected components. This class of component usually follows a non-uniform discretisation of available sizes across a predefined range. If real components are to be considered during optimisation then this discretisation must be represented.

10.3.2 Dealing with discrete elements

Discrete elements are those mechanical components where one or more of the attributes may only take a predefined range of values. If these attributes are assigned values that do not satisfy these bounds then a real component is not being considered and the solution produced is non-existent. In order to deal with discrete element values, the ability to search for an optimum solution from a predetermined list has been incorporated into the optimisation software. This function is called 'LVAR' (list variation) and allows both continuous and discrete variables to be considered concurrently by the optimiser. In addition to these constraints being applied to a particular component, they must also be applied when design parameters are selected for a component which will be directly coupled to a discrete component parameter. An example of this situation, is for the diameter of a shaft in the region where it is coupled to a bearing. Although in the shaft representation the diameter may be continuously variable, it will be restricted to discrete values for the internal diameter of the bearing. This discrete function must therefore be invoked for shaft models which are coupled to a bearing. This is made possible in the proposed approach because the macro or code necessary for optimisation is dynamically configured for each episode. It is therefore possible to evaluate the system representation for such dependencies and then incorporate this into the optimisation code.

---

[1] The term 'real' component refers to elements which may already exist, have been previously used and have predetermined or predefined properties. In many cases, these are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified.

10.3.3 Model resolution during optimisation

The modelling approach developed in this work represents the connectivity and types of component in a mechanical system, and from a set of performance requirements determines a system of real mechanical components from electronic representations. In order to achieve this, the system incorporates a strategy for data arbitration described in chapter 5. This ensures that more important design parameters such as those specified by the designer are propagated through the system model. This process aims to ensure that a real and compatible system of elements is determined. This approach must also be used during optimisation, so that successful solutions are pursued. To facilitate this, component attributes selected as design variables are assigned rankings of the highest precedence, equivalent to that specified by the designer. However, the approach developed in chapter 5 cannot independently resolve conflicts where parameters carry equal rankings of the highest precedence (designer specified). Consequently, it is important that the optimisation is not over constrained during problem formulation by driving related component attributes within component chains[2]. An example of this, is where the internal diameter of a bearing is driven by the optimiser as well as the diameter of the shaft node to which it is connected.

Another important consideration is to ensure that the performance envelope of a component type is not exceeded. This can be achieved by bounding particular component attributes within the optimiser. This prevents the optimiser using non-existent values for a component attribute from perhaps an electronic catalogue. In addition to this, the bounds checking implemented in the software modules that interface the various electronic representations can also be used to determine when limits have been exceeded for any attribute not driven by the optimiser. If the bounds for a particular attribute are exceeded then a failure is registered within the system. This approach can be extended for optimisation purposes so that a measure of failure is determined, which is equal to the value by which the bounds are exceeded. In this manner, the optimisation engine can acquire more information about the relative success or failure of exploratory moves.

10.3.4 A software architecture for optimisation

For the purpose of demonstrating the incorporation of optimisation techniques in the modelling approach, there is a need to create an intermediary software module between the modelling environment and the optimiser, and provide a mechanism for controlling the optimisation

[2] A component chain refers to a sequence of components that convey system inputs/outputs or link core elements.

episode. This control is necessary because both the optimiser and the modeller are independent software entities that require a finite length of time to execute their respective functions. To address these requirements, an intermediary software module is implemented to hold the necessary data for optimisation and suspend the processes in the modeller and optimiser as necessary. To achieve this, a Microsoft Excel Spreadsheet is used with the DDE (Dynamic Data Exchange) protocol, discussed in chapter 7. An overview of the functions of the software architecture is depicted in figure 10.3. The architecture can be separated into two distinct processes. The first is the formulation of the problem and the configuration of the optimisation macro and the second is the configuration of the intermediary spreadsheet. The optimisation macro is also constructed in a spreadsheet, in conjunction with a Visual Basic for Applications (VBA) code module. The objective function and the design variables are selected in the modelling environment, this information is then passed to the spreadsheet. Upon receipt of the information an intrinsic command is invoked which creates a customised macro for the optimiser. This macro is created by the spreadsheet from a generic code template that follows a similar format to that shown in figure 10.1. This code macro is saved as an input file for the optimiser, which is read in when optimisation is invoked.

The second part of the architecture involves the construction of the intermediary spreadsheet that controls the optimisation episode. The data elements held within this intermediary are depicted in figure 10.4. The intermediary is capable of handling any number of design variables, each is assigned an individual identifier and also carries the name of the component attribute and the identifier used in the modelling environment. Furthermore, the element value and desired initial step size is also conveyed. For the construction of the objective function all five possible terms are conveyed; cost, mass, and the leading dimensions in the x, y and z planes. However, only those which carry an inclusion flag of status '1' are built into the overall objective function. The other elements incorporated in the intermediary are necessary for control and error calculations. The status of the 'opt flag' is changed to '1' when the desired number of steps for the optimisation episode has been completed. The status of this flag is monitored by both the modeller and the optimiser, which both terminate their respective processes upon a change in status. The two 'control flags' are also monitored and accessed by the modeller and the optimiser respectively. This gives four possible states, only three of which are used. One state is used to activate the modeller, one to activate the optimiser and one is a neutral state which allows any current processes to be completed. In this manner, the order of operations for optimisation can be controlled. The final data elements held in the intermediary are necessary to denote instances when the modeller has failed to determine a solution state, and where this occurs, provide a measure of the failure. These are the 'system fail flag' and the 'error value' respectively.

10.4 Optimisation of an assembly

This section provides an overview of the procedure for setting up an optimisation episode within the modelling environment. The complete process for constructing the model, formulating the objective function and the subsequent optimisation process is depicted in figure 10.5. A schematic representation of the conceptual solution is constructed, the governing models for the components are specified and the performance requirements set. This phase of the process is identical to the normal model construction procedure described in chapter 8. Once the model has been fully configured an objective function and associated design variables must be selected. Following which, the optimisation code can be generated by the system. This procedure remotely activates the intermediary spreadsheet and the optimiser, which reads the appropriate macro. The first iteration in the optimisation cycle utilises the initial values for design variables supplied by the modeller. Throughout the process, communication between the modeller and the optimiser takes place through the intermediary until the desired number of iterations or convergence has occurred. All processes other than model construction (user operations) are automated by the modelling environment and require no user intervention.

This strategy has been implemented in the modelling environment and demonstrates that optimisation methods can be integarted into the overall modelling approach. The implementation of this strategy is not meant to be a robust architecture rather it merely demonstrates that through the process of interfacing various software tools it is possible to incorporate third party optimisation tools with the modelling approach. The reliable treatment and optimisation of the particular problem is beyond the scope of this body of research and as a consequence of this the case studies have not been optimised. However, the investigation of this problem provides an important area for future work.

10.5 Concluding remarks

This chapter has discussed and developed the issues associated with the incorporation of optimisation techniques into the integrated modelling approach. In particular, these issues relate to the complex solution space created by the inclusion of standard components and the incorporation of third party representations. The latter complicates an overall strategy because data is only ever implicitly defined within these representations and these independent representations create a largely nonholonomic system. The purpose of this phase of the research is not to develop an optimum implementation for the optimisation of the given problem, rather it is to identify the critical components necessary to provide for optimisation and develop a strategy to enable the application of optimisation techniques to an integrated modelling environment. The development of the key aspects of this strategy have been discussed and a software architecture has been created. This architecture enables a third party optimisation tool to be interfaced with the modelling environment such that no particular optimisation approach is prescribed. Preliminary trials with the optimiser have demonstrated that the correct sequence of operations and complete data transfer are possible at runtime. However, full optimisation episodes have not been undertaken and the ability of the strategy to provide for the reliable treatment of the problem has not been demonstrated. However the development of the issues and the creation of possible software architecture provide a basis for future work.

The development of methods that deals with the optimisation of systems represented within an integrated modelling environment is particularly important because the environment provides a solution space that bounds 'real' engineering components rather than approximations or feasible component sizes, which is the approach used in other research tools. If these approximate sizes are used for the purposes of optimisation, then ultimately during the detailed design phases, components may have to be resized (in order to procure a real component). If this resizing is necessary then other components may well have to be altered to accommodate the changed part. Furthermore, even for slight changes in component specifications, if the optimised solution is sensitive these changes may at best produce a sub-optimal solution or even a poor performing solution. The reliable treatment and solution of the particular problem is beyond the scope of this work and provides an important area for future work.

```
$Test.mac

dec real x, y, z;          Declare variables
x = 0.0;                   Initialise variables
y = 0.0;
z = 0.0;

function solve
{
  var x, z;                          Select parameters to vary
  lvar y, 1, 2, 3, 4;                Specify the discrete values for a parameter
  rule( x – y – z );                 Construct constraint rules
  rule( 2*x – y - 2*z – 2 );         Rule operator minimises the value of the function
  rule( 3*x – 2*y – 3*z – 1 );
}

solve();                   Call the solve function

fwriteln(0, "x =", x);     Write to the screen
fwriteln(0, "y =", y);
fwriteln(0, "z =", z);
```

Figure 10.1 – Example macro for the solution a set of equations using constraint rules

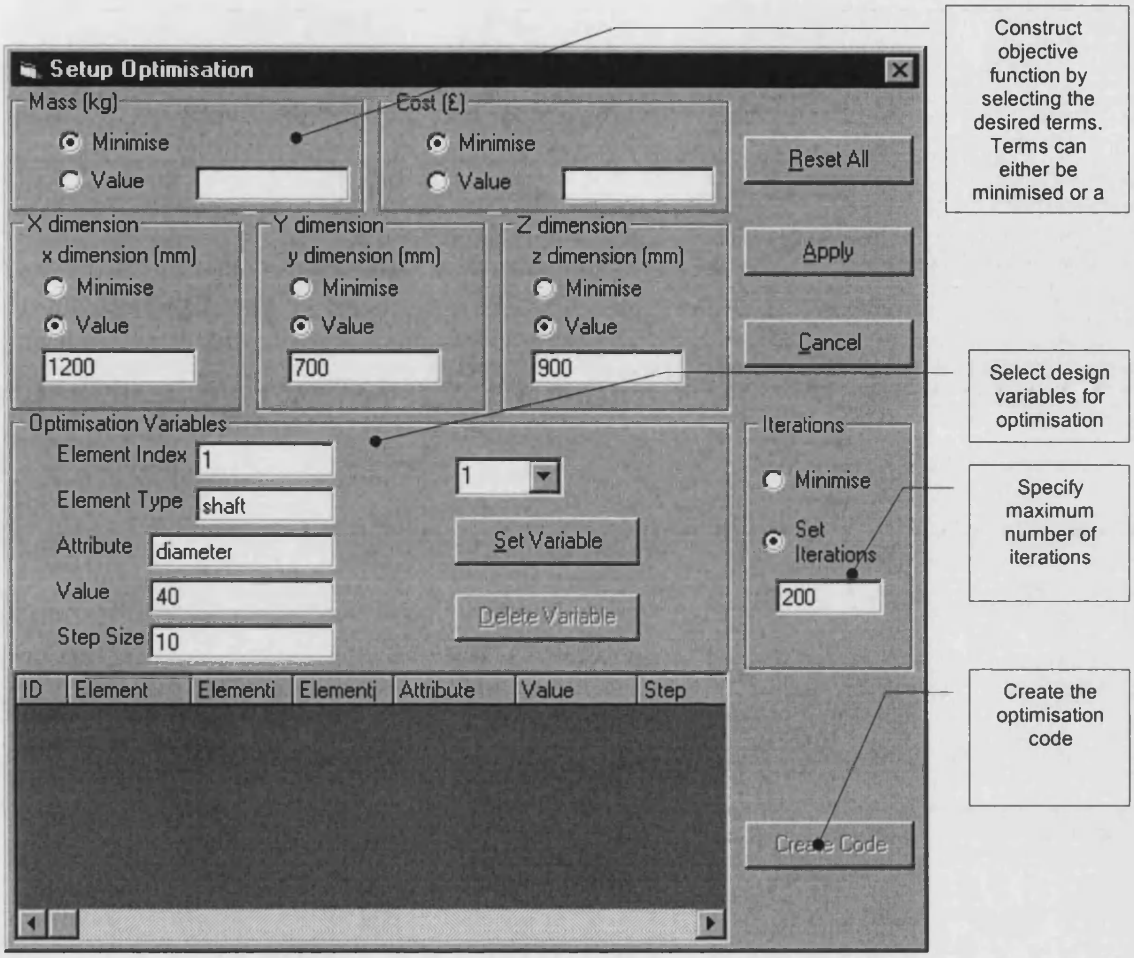


Figure 10.2 – Software interface for the construction of the objective function and selection of optimisation variables

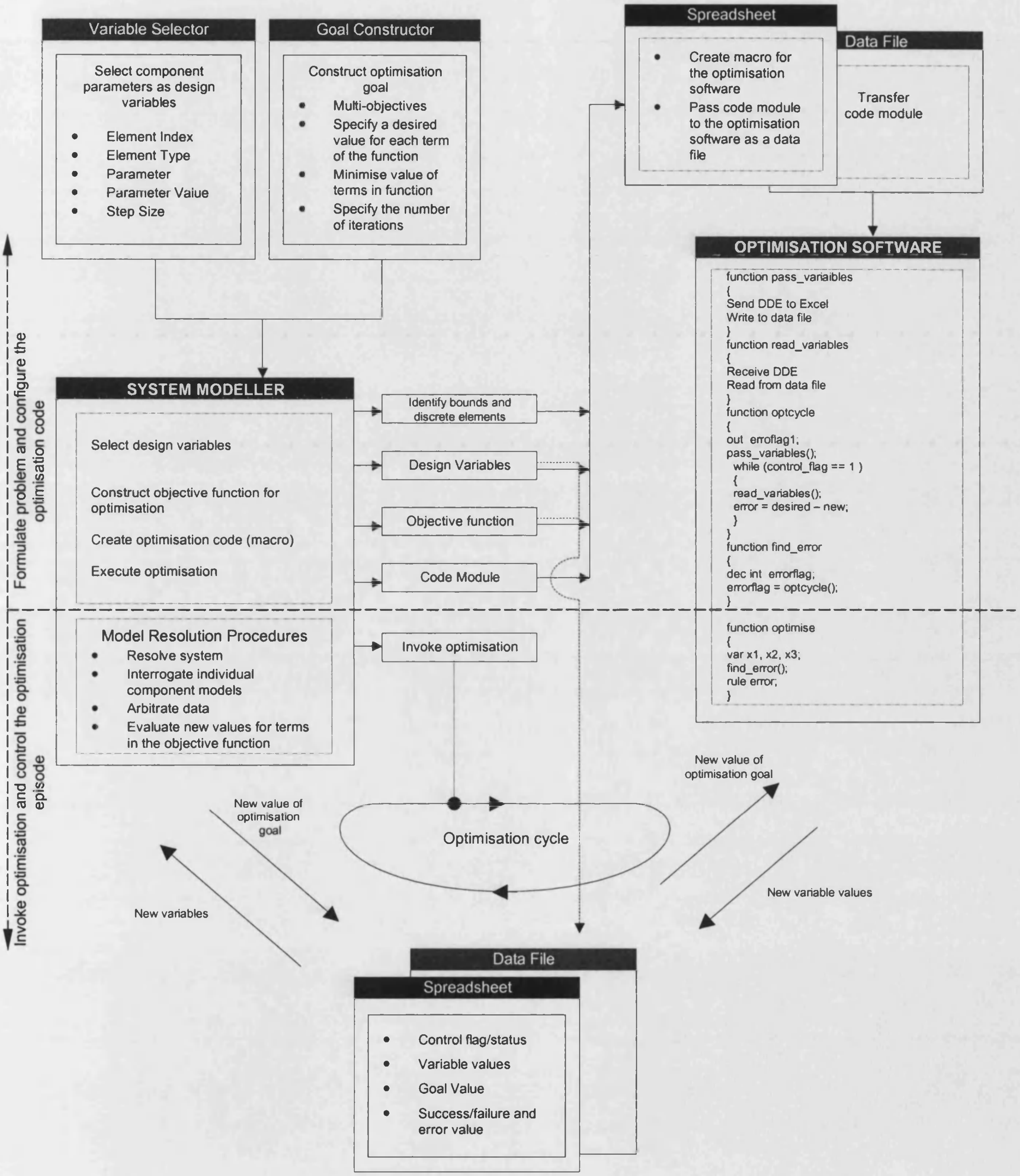


Figure 10.3 – Schematic overview of the software architecture for optimisation of a system model

Optimisation Goal Values				Control Operators	
	Actual	**Goal**	**Include**	**Control Flag1**	0
Mass	122	0	0	**Control Flag2**	0
Cost	2456	2000	1	**System Fail**	0
x	1321	1200	1	**Opt Flag**	0
y	723	600	1	**Error Value**	0
z	567	600	1		

Optimisation Variables				
No of variables	6			
ID	**Element**	**Attribute**	**Element Value**	**Step Size**
1	3	diameter	40	1
2	6	length	163	1
3	9	ratio	4	0.1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Figure 10.4 – Elements held in the intermediary software module during optimisation

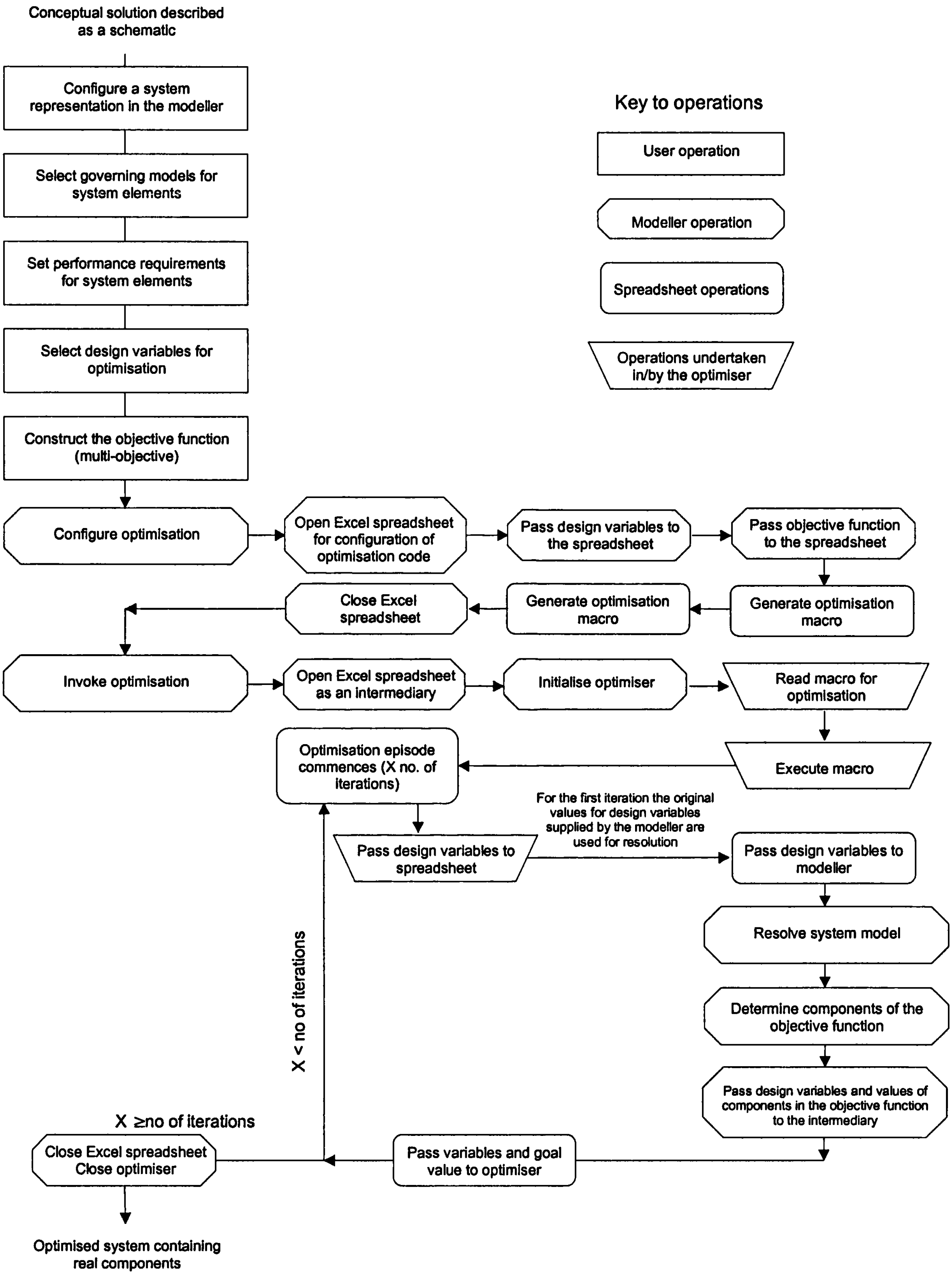


Figure 10.5 – Process diagram for the optimisation of a model in the integrated modelling environment

Chapter 11

Case studies

The previous chapters describe the development of a new modelling approach to support the design of machine systems. The approach provides support for the embodiment of a machine system with a set of real mechanical elements and in particular standard selected and standard designed components. The key aspects of this modelling approach are the strategy for system representation, the protocol for handling interactions, the resolution process, and strategies for data arbitration, compatibility analysis and the integration of third party electronic representations. Theses various elements of the modelling approach have been discussed in detail and demonstrated individually in the previous chapters. The incorporation of these elements in a computer based support tool for the early stages of the design process is described in chapter 8. The capabilities of this support tool are further enhanced by the development and introduction of cost modelling techniques for individual mechanical components, which provides an indication of the overall system cost.

In order to demonstrate the capabilities of the support tool and the feasibility of the overall modelling approach in its entirety the application of the software tool to a number of case studies is discussed. It is not the goal of this research work to create the optimum software implementation or to provide the most reliable method or approach for dealing with the specific problem, rather it aims to identify and develop the essential elements of the new modelling approach and demonstrate the feasibility of the overall strategy. Furthermore, the three case studies are chosen in order to demonstrate the capabilities and benefits of particular aspects of the modelling approach. The first of these case studies is a simple transmission, which aims to demonstrate the ability of the support tool, and hence the modelling approach, to handle and automatically embody design solutions with altered configurations and performance requirements. The second case study involves a more complex system with two subassemblies and demonstrates the ability of the approach to handle large systems and embody a solution to meet essential physical (spatial) constraints. The final case study is a relatively complex transmission system taken from an industrial overwrapper. This case example includes multiple subassemblies and demonstrates the ability of the approach to evaluate the effect of changes to an individual assembly on the overall system and in particular cost, mass and spatial occupancy.

11.1 Case study 1 – simple transmission

The first case study involves a relatively simple transmission, comprising a single layshaft that connects two chain drives and is supported by two bearings. Such transmissions are commonplace for achieving either a reduction in speed, connecting a power source and load, or transmitting power around an obstacle or restricted space. For the case considered in this work, the requirements of the transmission are to provide a 1:1 speed ratio and connect the motor and load. Although the motor and the load are positioned on a common centre, the layshaft assembly is used to avoid another mechanism. To achieve this, an envelope of 150 mm by 350 mm with appropriate clearances must be incorporated in the transmission design. A concept sketch for the topology of the transmission is shown in part (a) of figure 11.1. This sketch is used to create the schematic or connectivity diagram of the transmission in the modelling environment, shown in part (b) of figure 11.1. The spatial constraints are embodied in the system model by specifying the axial length for sections of the shaft and the centre distances for the chain drive components. Furthermore, the power and speed constraints for the system input (motor) and the output (load) are entered. The system model is then resolved. This resolution process culminates with a system of fully specified components which are physically connectable and matched in terms of their performance capabilities. For each component a full set of attributes is determined. Figure 11.2 shows the attributes for one of the bearings and one of the chain drives. Using this real component data enables a three-dimensional representation of the embodied solution to be constructed. This provides a mechanism for the verification of the resolved system and is illustrated in part (a) of figure 11.3. Furthermore, the real component data and the inclusion of cost modelling techniques enable the generation of a mass value, cost value and spatial occupancy for the system considered. The attributes of the embodied design solution, including overall cost, overall mass and spatial occupancy, as well as the cost and mass of each component, are displayed in the 'System Information' window, shown in part (b) of figure 11.3.

The primary objective of this first case study is to demonstrate the ability of the modelling approach to evaluate a system with different performance requirements and in particular different power ratings. To achieve this, the system is resolved for a number of instances with different power ratings. These are shown in figures 11.3 to 11.7. The results produced by each modelling episode can be used to generate system characteristics over different power ratings. These characteristics are shown in figure 11.8, parts (a) to (c) and figure 11.9, parts (a) and (b). The characteristics include cost, mass, spatial occupancy, power-to-weight ratio and cost-weight ratio. Such information provides a useful insight into the relative changes in the system attributes necessary to deliver the different power requirements. For example, part (c) of figure 11.8 shows that no significant increase in spatial occupancy is necessary for the transition from 2 kW to 4 kW or from 8kW to 10 kW. Such information may be particularly useful where a number of transmissions variants, with different capabilities, are to be manufactured and spatial constraints are critical. Furthermore, it is possible to investigate the variations in the attributes of individual components for various different system configurations. For example, figure 11.10 shows the variation in the attributes of one of the bearings in this case study.

The first example is deliberately chosen to be simple in order to illustrate the approach. The model can easily be modified to handle components located on opposing sides of the layshaft, shown in figure 11.11, or additional requirements, such as providing a 4:1 reduction in speed, illustrated in figure 11.12.

11.2 Case study 2 – dual assembly transmission

The second case study involves a more complex transmission system that possesses two layshafts. Such a system can be thought of as comprising two distinct assemblies, each centred around one of the layshafts. This case is used to demonstrate the ability of the modelling approach to handle large systems containing more than one assembly/subassembly, where each assembly is denoted by a core component, as defined in chapter 4. For the purpose of this work, the case study also requires that a number of performance and physical requirements are achieved. In particular, the system must span a 600 mm gap between the prime mover and the load, using two layshafts running at different speeds. Furthermore, the transmission must provide an overall speed ratio of 1:1, across two sets of gears and a chain drive.

A conceptual sketch of the transmission is depicted in part (a) of figure 11.13. A schematic or connectivity diagram of the system is created in the modelling environment, shown in part (b) of figure 11.13. For the purposes of comparison and to demonstrate the incorporation of the spatial and performance constraints, the system is firstly resolved with a set of default requirements for both the dimensions of the shafts and the gear pairs. The geometry and attributes of the resolved system are shown in parts (a) and (b) of figure 11.14 respectively. The system is then resolved again with the additional physical constraints imposed on the centre distance of the chain drive component, the length of the shaft sections; necessary in order to comply with the 300mm clearance for the upper layshaft (see figure 11.13), and the shaft separation for the gear pair. The resolved system satisfies all the imposed performance and physical constraints. The geometry of the embodied solution and the system attributes are shown in parts (a) and (b) of figure 11.15 respectively.

This case study comprises nineteen elements and two assemblies, compared to eleven elements and a single assembly in the previous case study. The ability of the modelling approach to represent and analyse systems comprising multiple assemblies overcomes one of the main limitations of previous work. This limitation relates to the fact that many modelling approaches and their associated computer based tools restrict the size and structure of the system to be modelled, and hence severely restrict the designer. This is discussed in detail in chapters 2 and 3.

11.3 Case study 3 – drive train for an overwrapper

The third case study involves the reconfiguration of a drive train for an industrial overwrapper, shown in figure 11.16. The overwrapper uses two cam and linkage assemblies to perform the wrapping of the film around the product. These assemblies are mechanically coupled by the drive train. The current configuration comprises two layshafts located 500mm to either side of a drive shaft, which is powered by an electric motor that is geared to the drive shaft. The layshafts are each driven by identical chain drives, shown in part (a) of figure 11.17.

The company for whom the case study has been done are undertaking a program of redesign and wish to alter some the machine assemblies that perform the gluing and tucking operations. In order to achieve this, additional space in the central/upper portion of the machine must be created. Furthermore, in an effort to reduce the out-of–balance forces it is desirable for the cams to rotate in opposite senses. In order to achieve this, a solution is proposed which involves shifting the drive shaft towards the lower layshaft and replacing the chain drive with a gear pair, shown in part (a) of 11.20. This increases the free space in the central/upper portion of the machine, allowing the redesigned assemblies which undertake the secondary production operations to be incorporated into the machine.

In order to investigate the reconfigured drive train, modelling episodes for the current machine configuration and the reconfigured layout are undertaken. A schematic or connectivity model of the current machine configuration created in the modelling environment is depicted in part (b) of figure 11.17. The required values for the centre distances of the chain drives, the loading and the shaft dimensions are entered. The system model in then resolved and a set of components determined. The system geometry is shown in figure 11.18 and the system attributes are shown in figure 11.19. This process is repeated for the reconfigured drive train, except one of the chain drive components is replaced by a gear pair and the centre distance for the upper chain drive and the gear pair are altered to 750mm and 250 mm respectively. The schematic, system geometry and system attributes are shown in figures 11.20, 11.21 and 11.22 respectively.

The production of system geometry and the generation of system attributes provide a means for comparing the revised layout and the original, these are summarised in figure 11.23. In particular, the mass of the system is reduced slightly as well as cost. This is primarily due to the introduction of the second gear pair, which reduces the forces in the lower layshaft. This enables a slightly reduced specification for the shaft and the bearings. Furthermore, the gear pair is slightly cheaper than the current chain drive assembly. However, the cost of the chain drive on the upper layshaft has risen slightly due to an increase in the centre distance and hence the chain length. The overall spatial envelope of the assembly has not changed, although there is a slight increase in the y-dimension due to the fact that the gear that has replaced the sprocket on the lower layshaft has a slightly larger diameter. This approach shows that there are no significant changes in the overall system attributes and that the redesign is a feasible solution. Furthermore, the effects on the system geometry can be further investigated by inspecting the system geometry. The system geometry is represented by 'simple solids'. This three-dimensional representation provides a mechanism for evaluating aspects such as clearances, arrangement, internal space and mounting points. An isometric view of the reconfigured system is shown in figure 11.24.

This case study demonstrates the capability of the modelling approach to evaluate the effect of changes to particular assemblies or components on the overall machine system. For the case study considered, the geometry of the existing system and the reconfigured system are shown in figures 11.18 and 11.21, and provide an indication of the changes to the internal space of the machine system. In addition to this, the modelling tool also provides an indication of the changes in system mass and cost which are brought about by incorporating the altered assembly.

11.4 Concluding remarks

The case studies have been chosen to illustrate and validate the various capabilities of the new modelling approach, and in particular:

1 Handle and automatically embody a design configuration for different performance requirements.

2 Automatically embody different configurations or design concepts for the same or similar performance requirements.

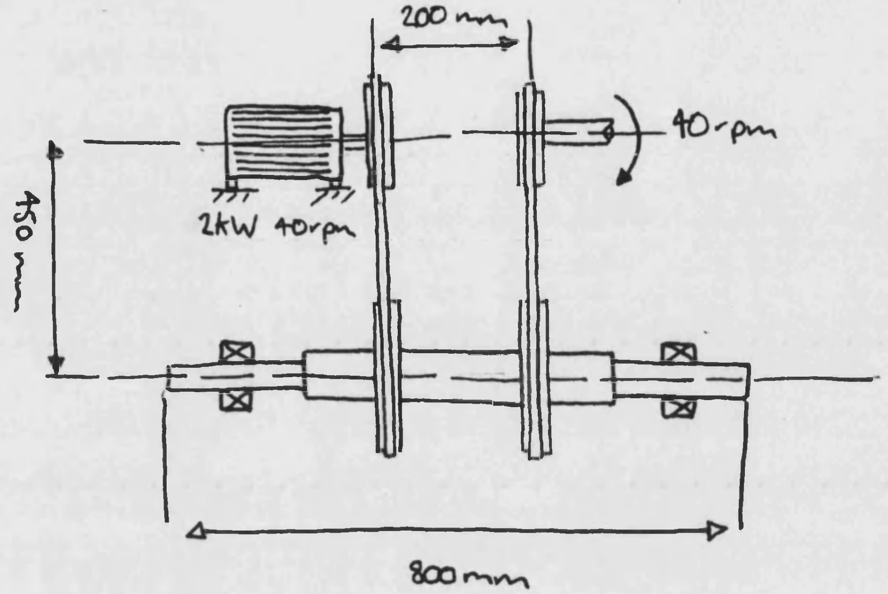
3 Assist the embodiment of systems to meet essential physical/spatial requirements.

4 Represent and handle large complex systems comprising multiple assemblies or subassemblies (case study 3 comprises three distinct assemblies). The first case study comprises a single assembly with eleven elements, the second consists of two assemblies and nineteen components, whilst the third case study comprises three assemblies and twenty eight components.

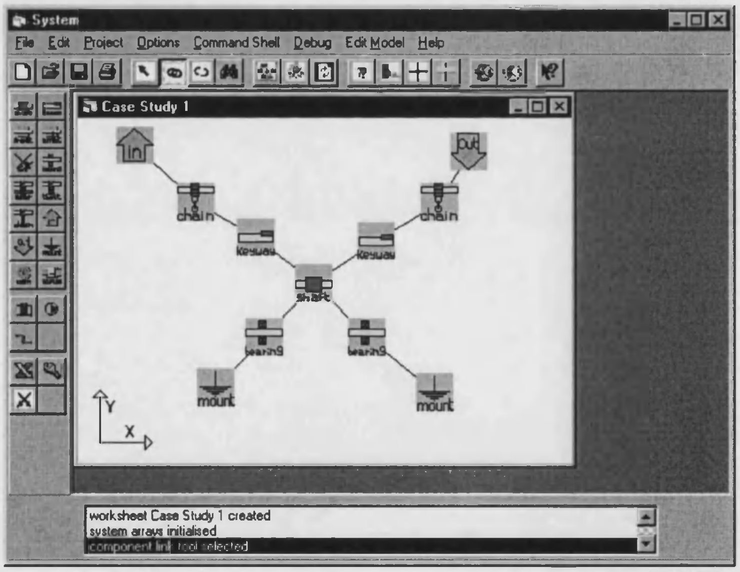
5 Evaluate the effects upon the overall system which are brought about by the introduction of a changed part or redesigned/reconfigured assembly.

The application of the software support tool to the three case studies demonstrates how the approach can be used to support the conceptual and embodiment phases of the design process. In particular, the approach enables the representation of concepts and assists their embodiment with a set of 'real' mechanical components to satisfy essential performance and physical constraints. The successful application of the modelling approach to the case studies, demonstrates the feasibility of the overall modelling approach and validates the hypotheses set out in this body of research.

The application of the modelling environment to the case studies also aims to demonstrate the overall aim of the new modelling approach. This involves the ability of the approach to support the designer in configuring, embodying and analysing design solutions comprising standard components. During these tasks the designer must consider a large number of components and consider an extensive range of attributes. For example, the third case study comprises twenty eight components, each of which possesses between ten and twenty four different attributes. If a system of this size were embodied manually by the designer, the process could take many hours, involving a number of analytically intensive and error prone tasks. However, the modelling approach is capable of handling and analysing such a system in only a fraction of the time. Thereby reducing the time taken to embody a design solution and allowing the designer to evaluate many more design alternatives. This enables the development of a more refined design solution and ultimately more fully informed decisions to be taken at an early stage in the design process. The inclusion of 'real' components in the modelling process ensures that the design solution comprises components that can be procured exactly as specified. This overcomes another of the limitations of many of the current modelling approaches which deal with abstracted models of components and therefore require that best or closest matched components be selected after the modelling phase.



Part (a) Concept sketch



Part (b) Concept schematic constructed in the modelling environment

Figure 11.1 – Case study 1 simple transmission

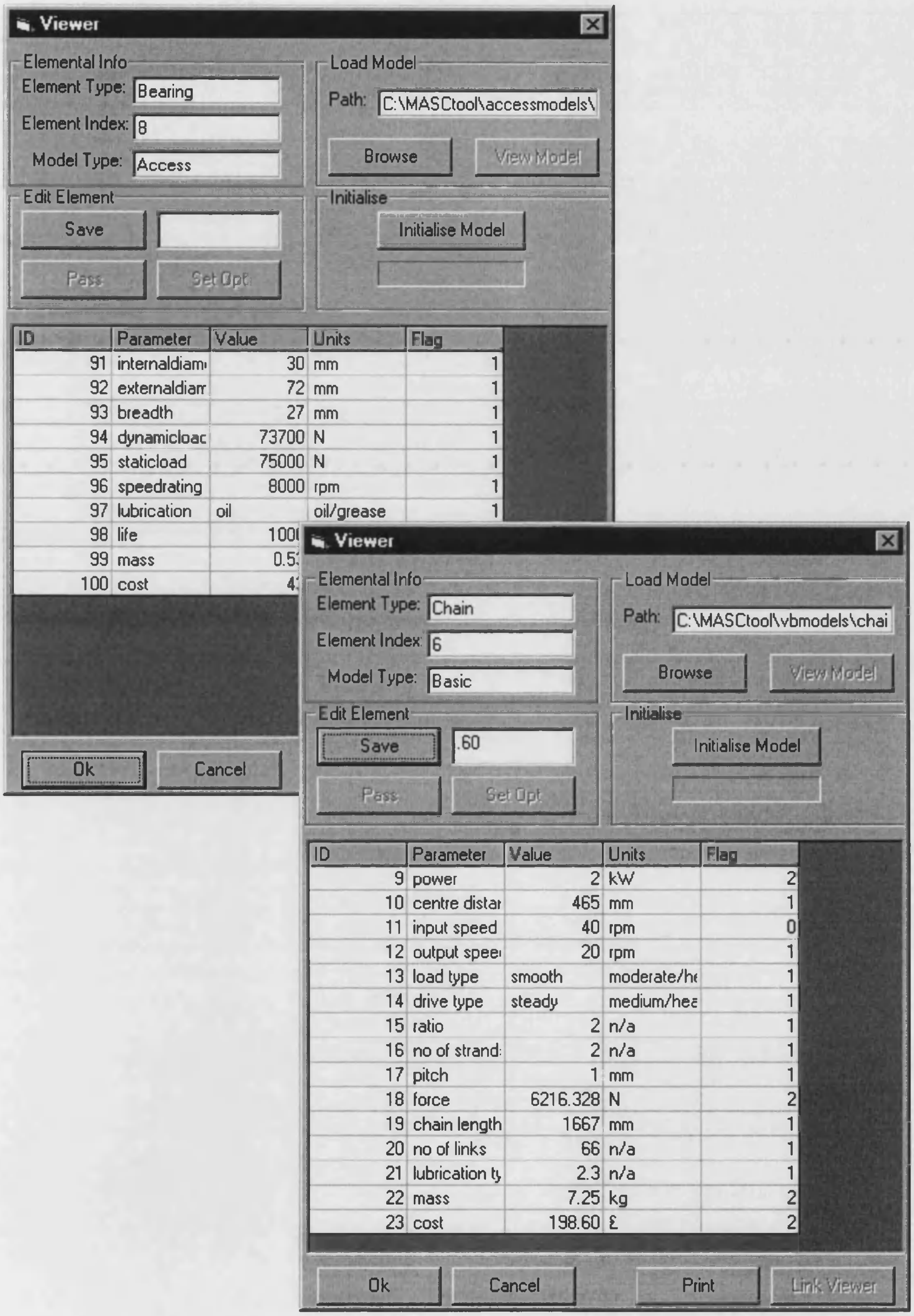
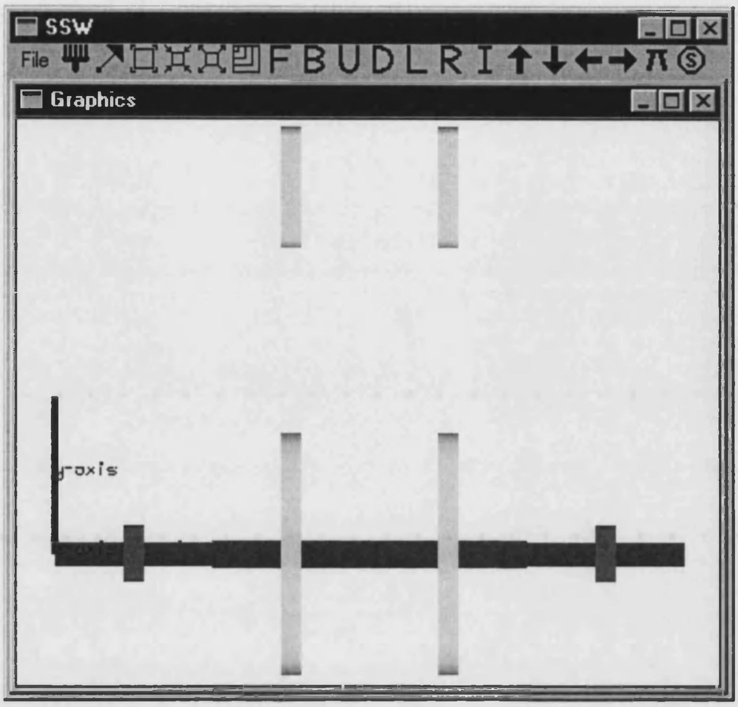
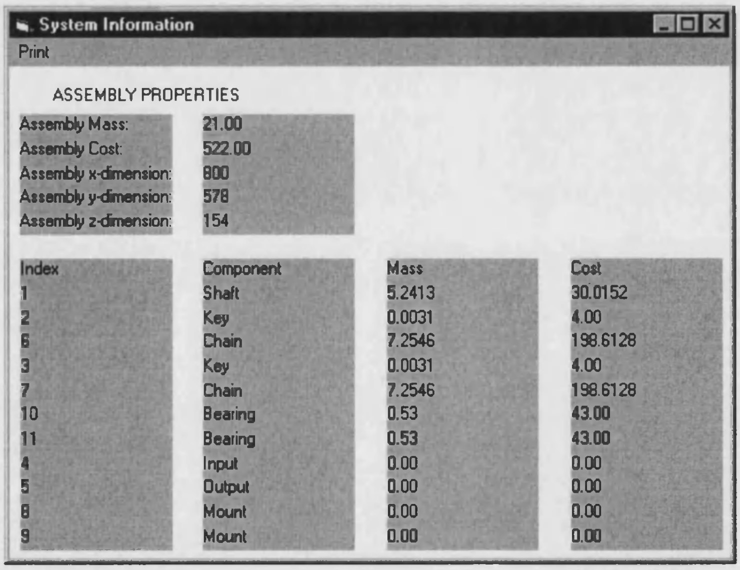


Figure 11.2 – Component attributes for a bearing and a chain drive



Part (a) System geometry



System Information

Print

ASSEMBLY PROPERTIES

Assembly Mass:	21.00
Assembly Cost:	522.00
Assembly x-dimension:	800
Assembly y-dimension:	578
Assembly z-dimension:	154

Index	Component	Mass	Cost
1	Shaft	5.2413	30.0152
2	Key	0.0031	4.00
6	Chain	7.2546	198.6128
3	Key	0.0031	4.00
7	Chain	7.2546	198.6128
10	Bearing	0.53	43.00
11	Bearing	0.53	43.00
4	Input	0.00	0.00
5	Output	0.00	0.00
8	Mount	0.00	0.00
9	Mount	0.00	0.00

Part (b) System attributes

Figure 11.3 – Case study 1 (2kW power rating)

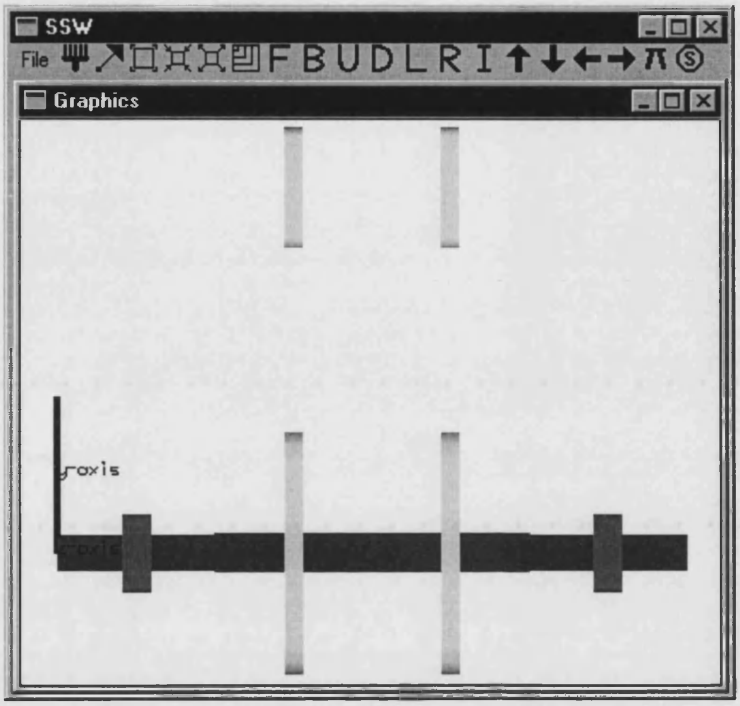


Figure 11.4 – Case study 1 (4kW power rating)

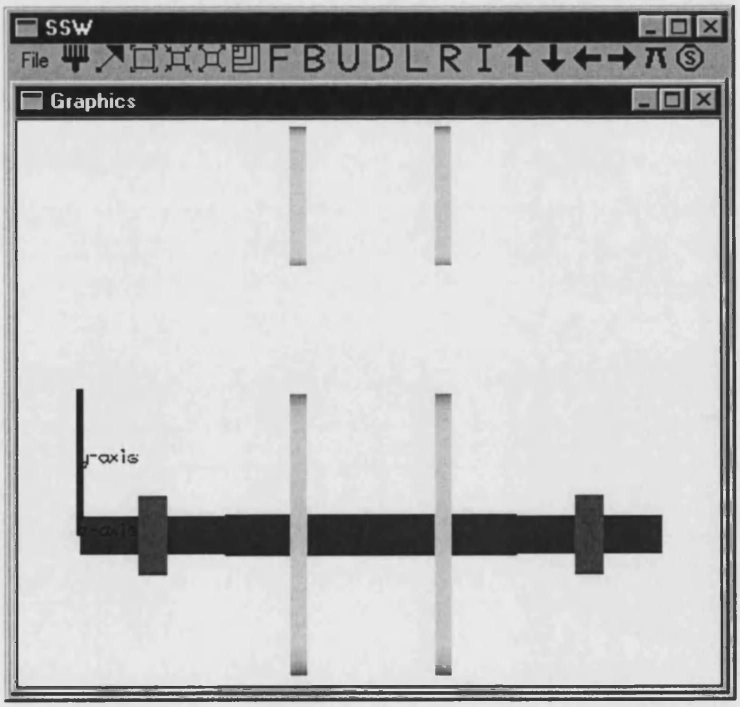


Figure 11.5 – Case study 1 (6kW power rating)

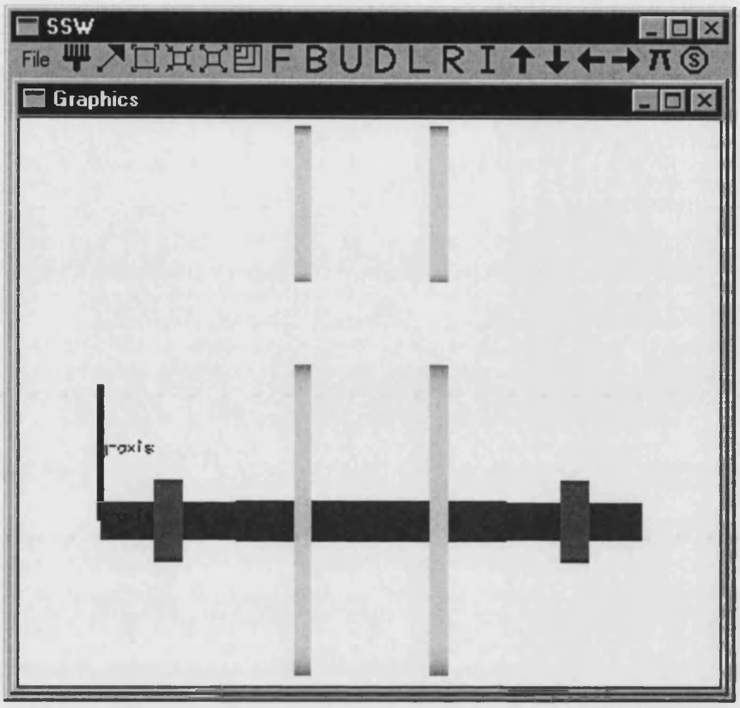


Figure 11.6 – Case study 1 (8kW power rating)

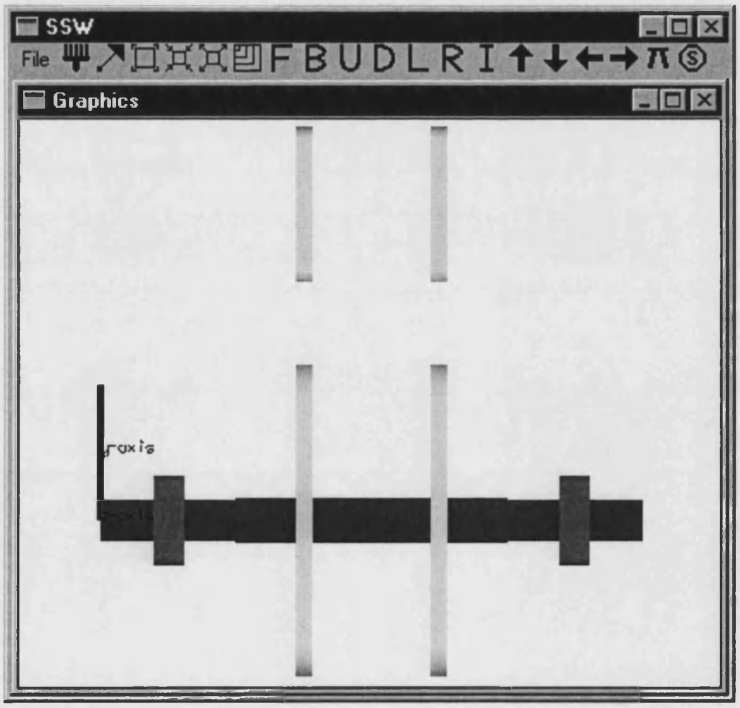
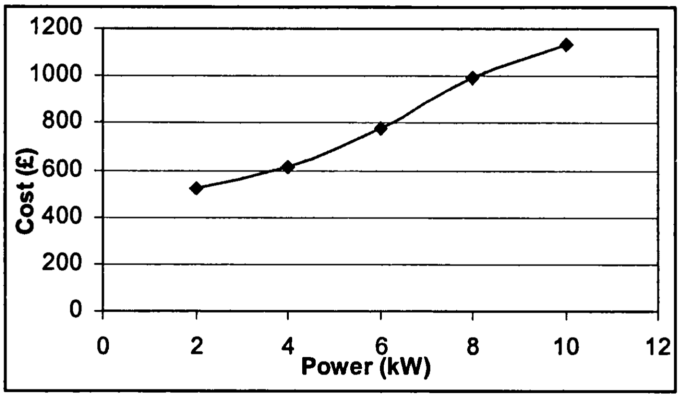
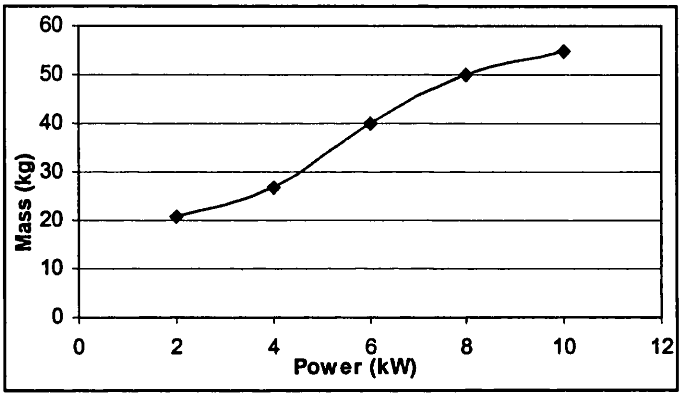


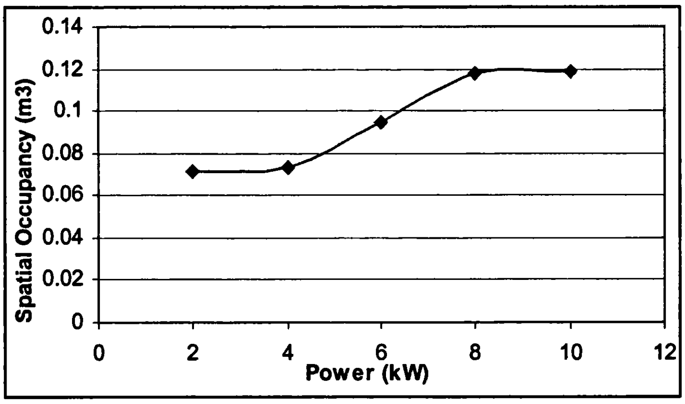
Figure 11.7 – Case study 1 (10kW power rating)



Part (a) Power against cost

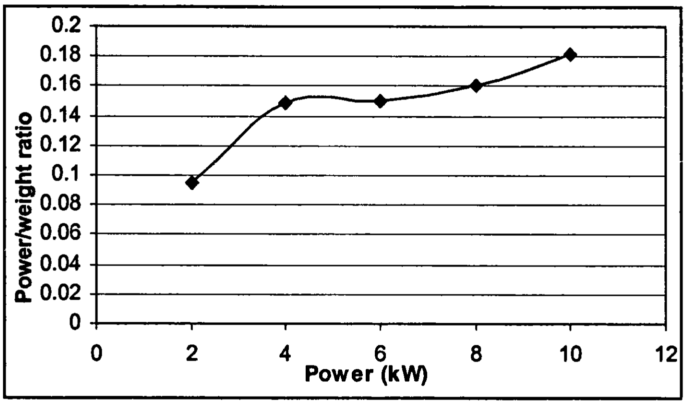


Part (b) Power against mass

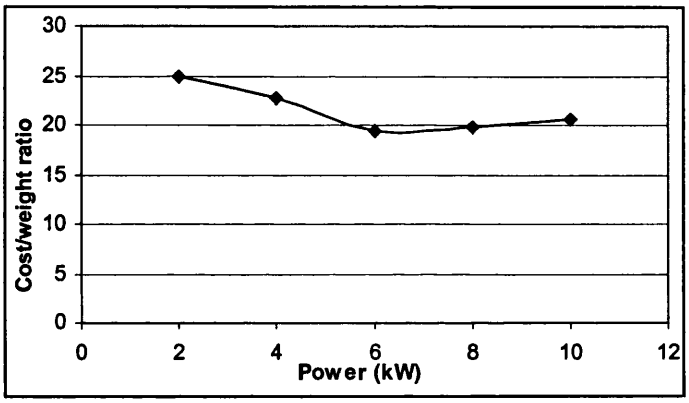


Part (c) Power against spatial occupancy (m^3)

Figure 11.8 – System attributes for different power requirements

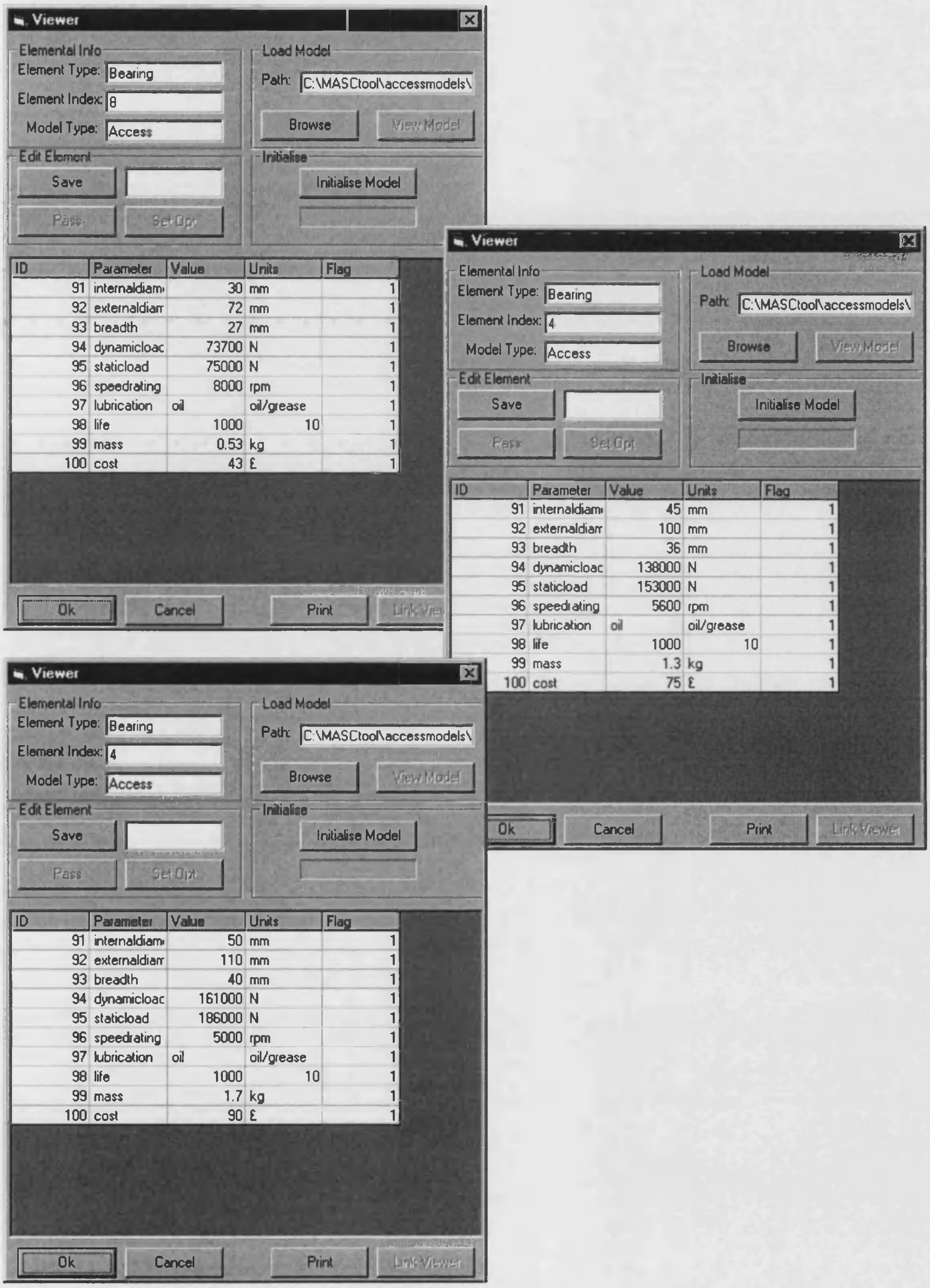


Part (a) Power to weight ration for different power requirements



Part (b) Cost to weight ration for different power requirements

Figure 11.9 – Attribute ratios for different power requirements

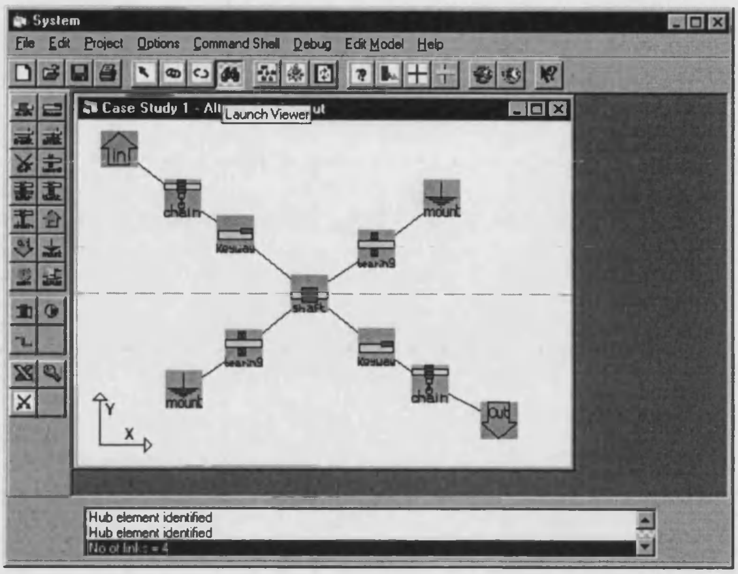


ID	Parameter	Value	Units	Flag
91	internaldiam	30	mm	1
92	externaldiam	72	mm	1
93	breadth	27	mm	1
94	dynamicloac	73700	N	1
95	staticload	75000	N	1
96	speedrating	8000	rpm	1
97	lubrication	oil	oil/grease	1
98	life	1000	10	1
99	mass	0.53	kg	1
100	cost	43	£	1

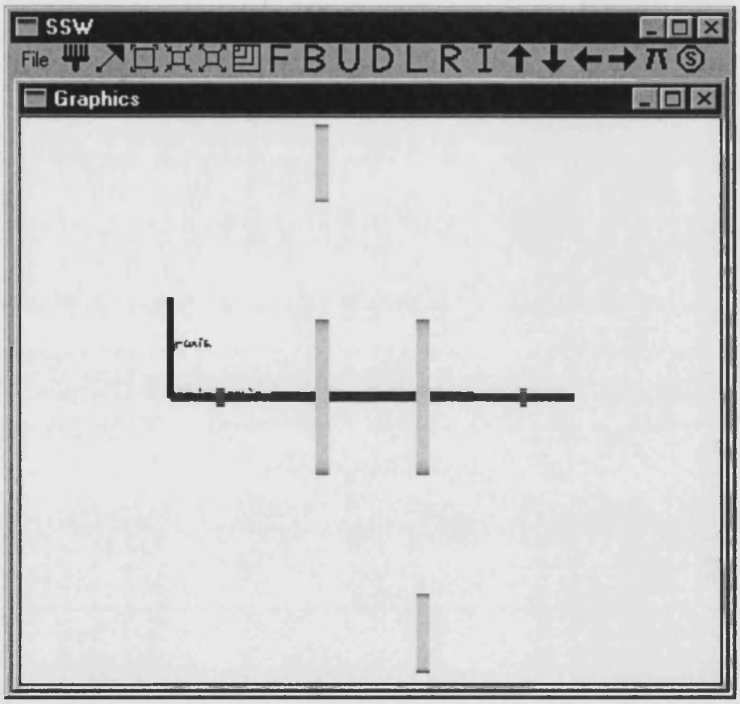
ID	Parameter	Value	Units	Flag
91	internaldiam	45	mm	1
92	externaldiam	100	mm	1
93	breadth	36	mm	1
94	dynamicloac	138000	N	1
95	staticload	153000	N	1
96	speedrating	5600	rpm	1
97	lubrication	oil	oil/grease	1
98	life	1000	10	1
99	mass	1.3	kg	1
100	cost	75	£	1

ID	Parameter	Value	Units	Flag
91	internaldiam	50	mm	1
92	externaldiam	110	mm	1
93	breadth	40	mm	1
94	dynamicloac	161000	N	1
95	staticload	186000	N	1
96	speedrating	5000	rpm	1
97	lubrication	oil	oil/grease	1
98	life	1000	10	1
99	mass	1.7	kg	1
100	cost	90	£	1

Figure 11.10 – Variation of bearing attributes for different power requirements



Part (a) Schematic for case study 1 with altered configuration



Part (b) System geometry for case study 1 with altered configuration

Figure 11.11 – Case study 1 with altered configuration (2kW power rating)

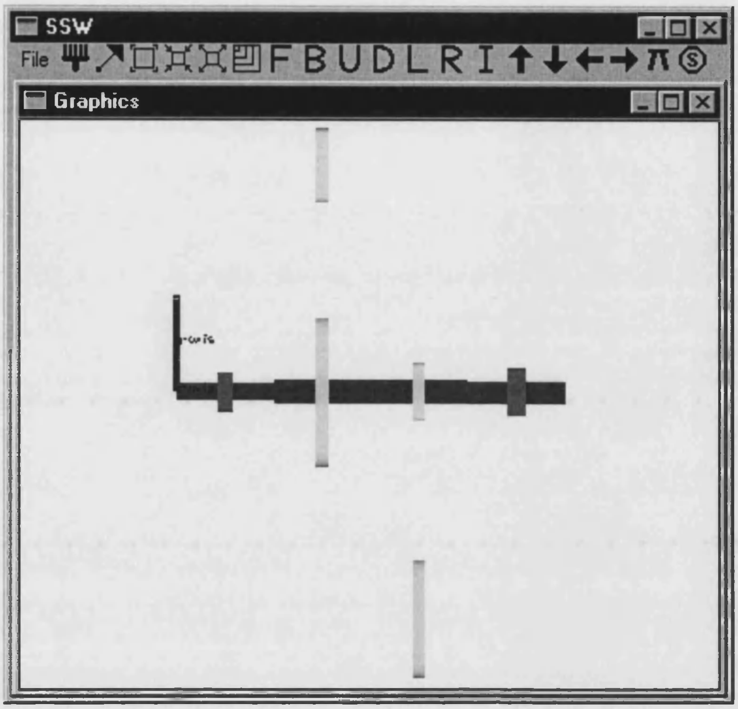
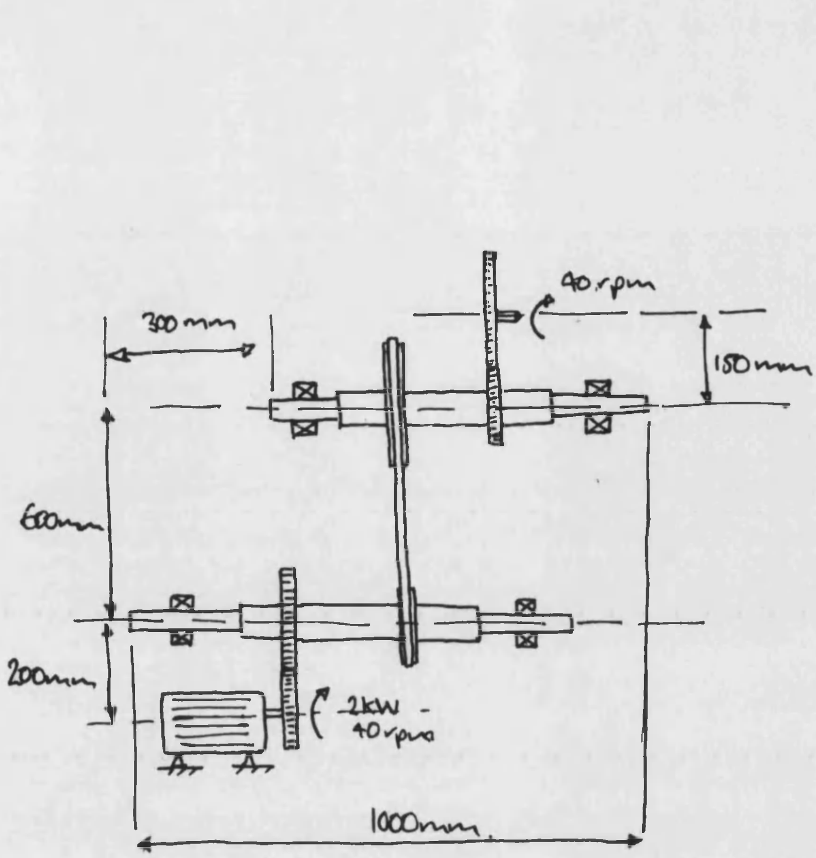
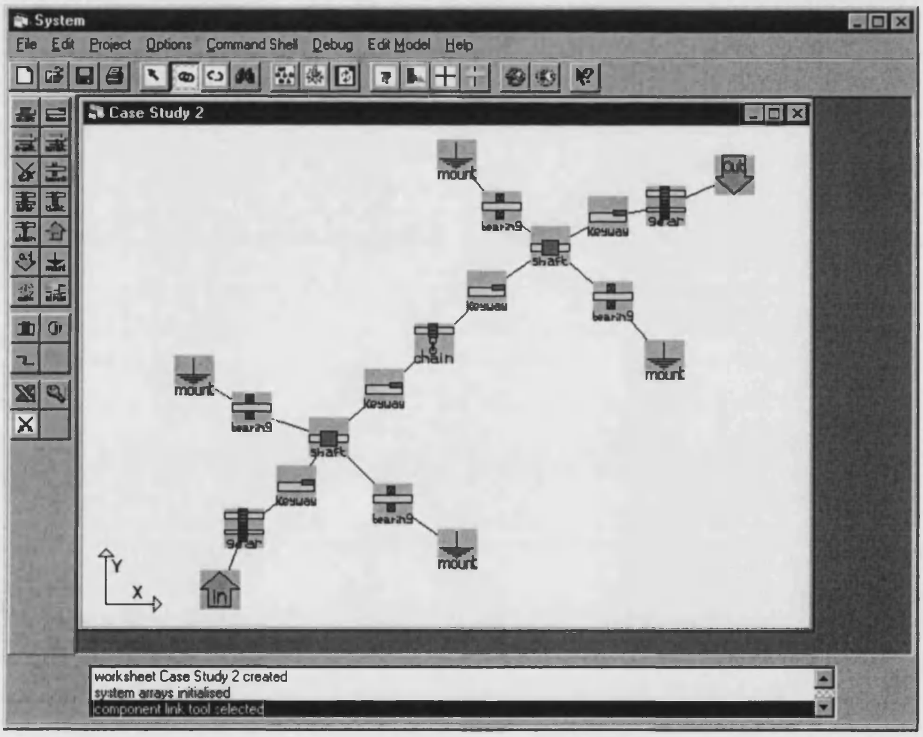


Figure 11.12 – Case study 1 with altered configuration and additional constraint for an overall reduction ratio of 4:1 (2kW power rating)

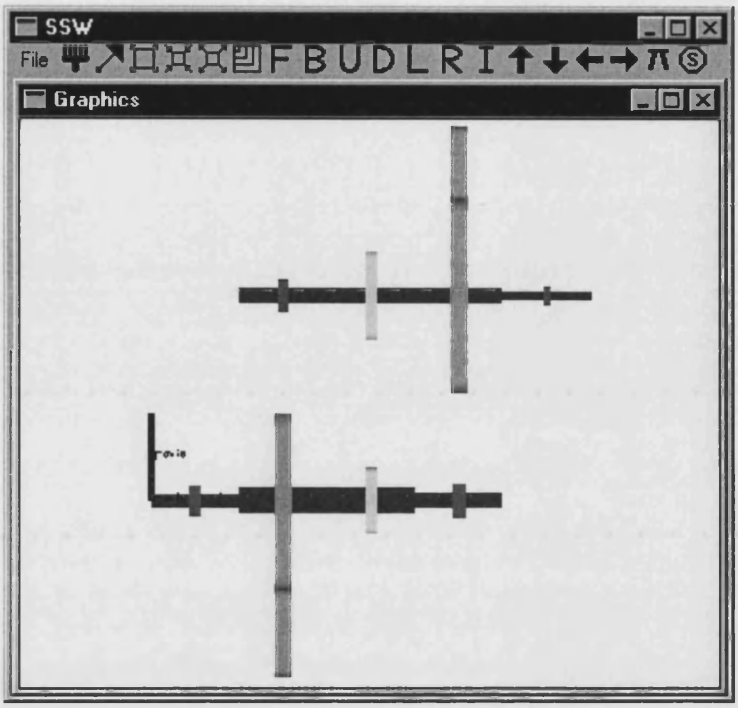


Part (a) Concept sketch



Part (b) Concept schematic constructed in the modelling environment

Figure 11.13 – Case study 2 dual assembly transmission (model configuration)



Part (b) System geometry

System Information

Print

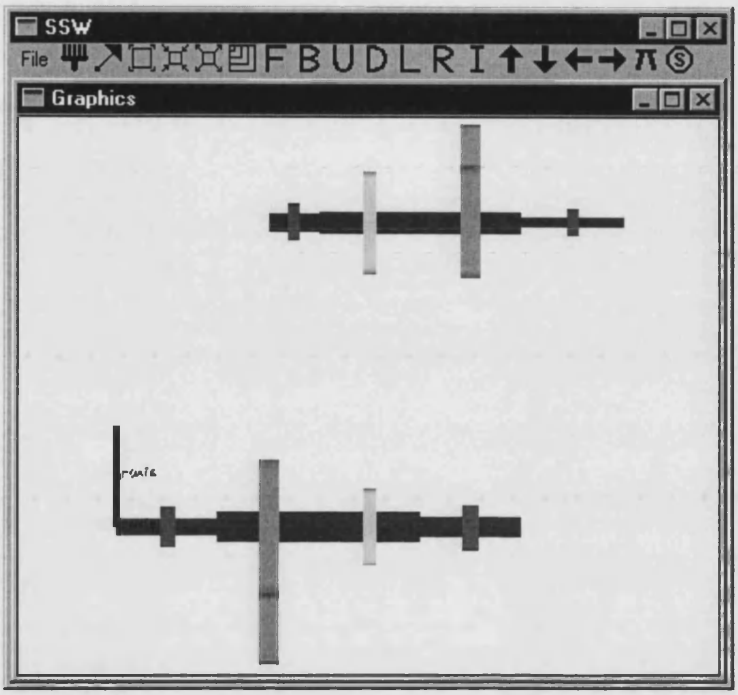
ASSEMBLY PROPERTIES

Assembly Mass:	41.00
Assembly Cost:	1098.00
Assembly x-dimension:	1000
Assembly y-dimension:	888
Assembly z-dimension:	202

Index	Component	Mass	Cost
1	Shaft	9.4264	51.6633
17	Key	0.0015	4.00
15	Chain	7.7505	193.2627
18	Key	0.0396	13.00
2	Shaft	8.1418	44.3865
8	Bearing	0.21	39.00
16	Key	0.03	5.00
13	Gear	6.1363	111.8193
9	Bearing	0.72	51.00
10	Bearing	0.65	43.00
11	Bearing	0.35	35.00
19	Key	0.0417	5.00
14	Gear	8.3999	501.8108
6	Mount	0.00	0.00
5	Input	0.00	0.00
7	Mount	0.00	0.00
3	Mount	0.00	0.00
4	Mount	0.00	0.00
12	Output	0.00	0.00

Part (b) System attributes and component information

Figure 11.14 – Case study 2 dual assembly transmission (system attributes)



Part (a) System geometry

System Information

Print

ASSEMBLY PROPERTIES

Assembly Mass:	53.00
Assembly Cost:	1383.00
Assembly x-dimension:	1000
Assembly y-dimension:	832
Assembly z-dimension:	202

Index	Component	Mass	Cost
1	Shaft	9.4577	51.8182
17	Key	0.0313	6.00
15	Chain	9.1295	202.7806
18	Key	0.0398	13.00
2	Shaft	7.1552	38.958
8	Bearing	0.21	39.00
16	Key	0.0311	6.00
13	Gear	15.1362	387.689
9	Bearing	0.72	51.00
10	Bearing	0.94	61.00
11	Bearing	0.16	26.00
19	Key	0.0418	5.00
14	Gear	10.5499	493.9791
6	Mount	0.00	0.00
5	Input	0.00	0.00
7	Mount	0.00	0.00
3	Mount	0.00	0.00
4	Mount	0.00	0.00
12	Output	0.00	0.00

Part (b) System attributes and component information

Figure 11.15 – Case study 2 dual assembly transmission with physical constraints

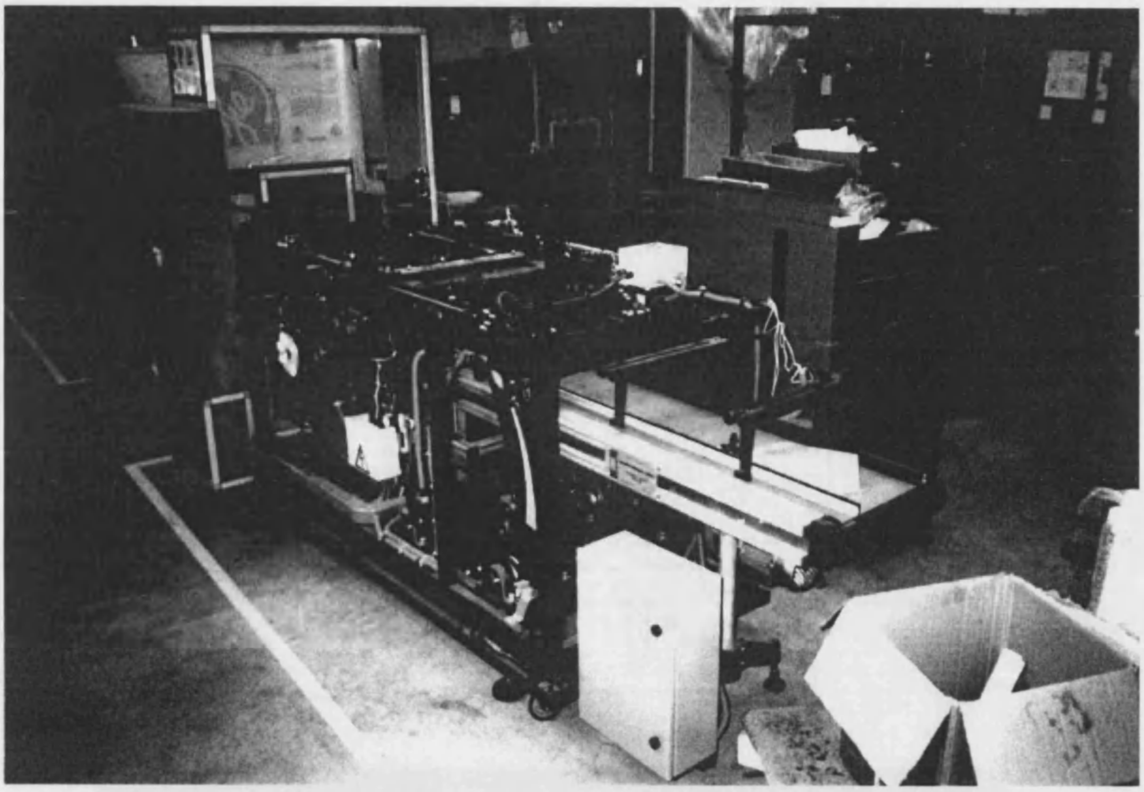
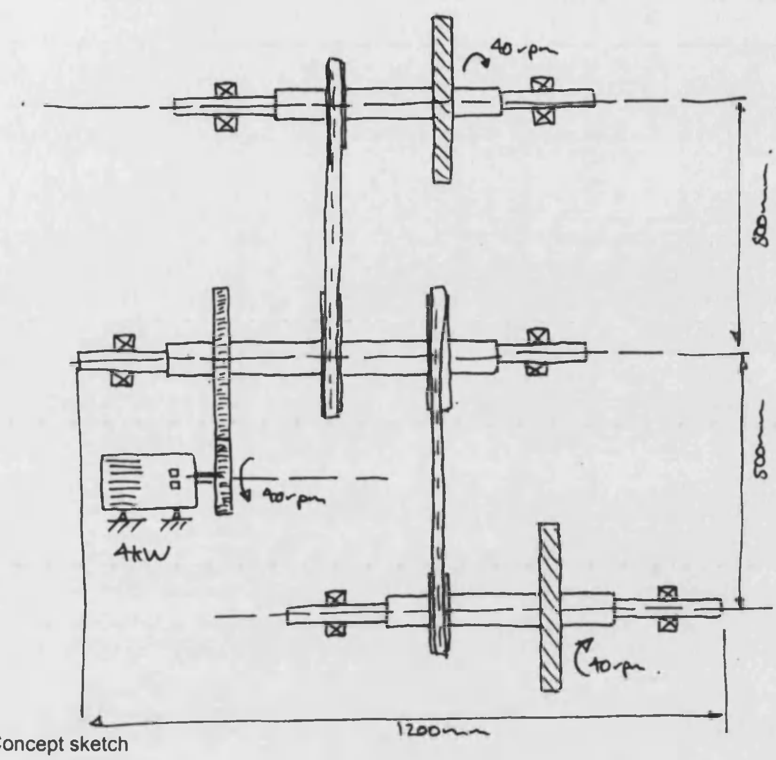
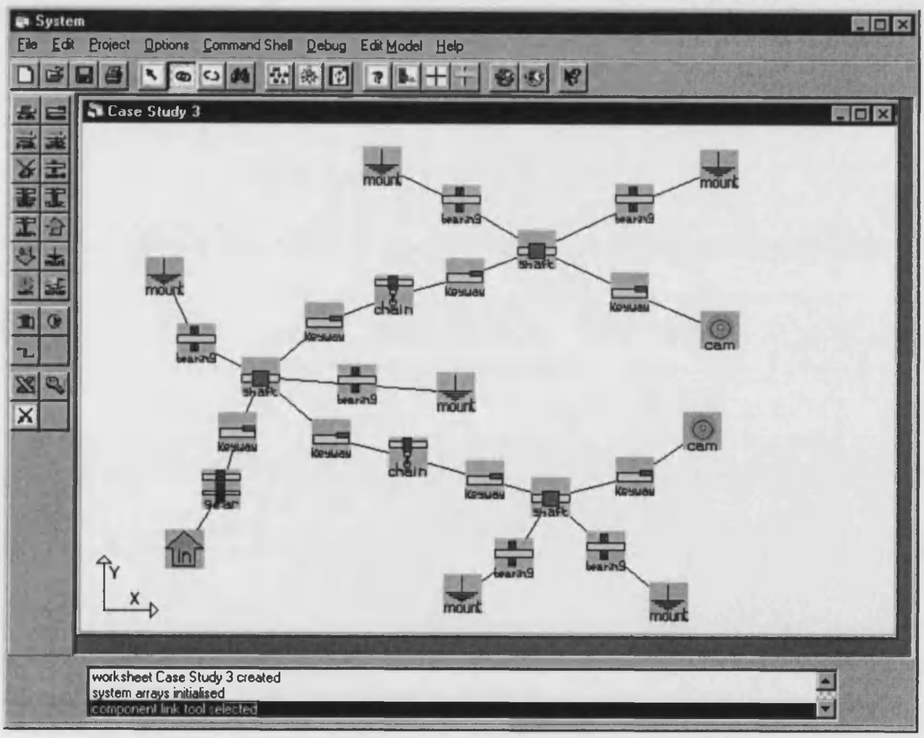


Figure 11.16 – Case study 3 industrial overwrapper



Part (a) Concept sketch



Part (b) Concept schematic constructed in the modelling environment

Figure 11.17 – Case study 3 current transmission for the overwrapper

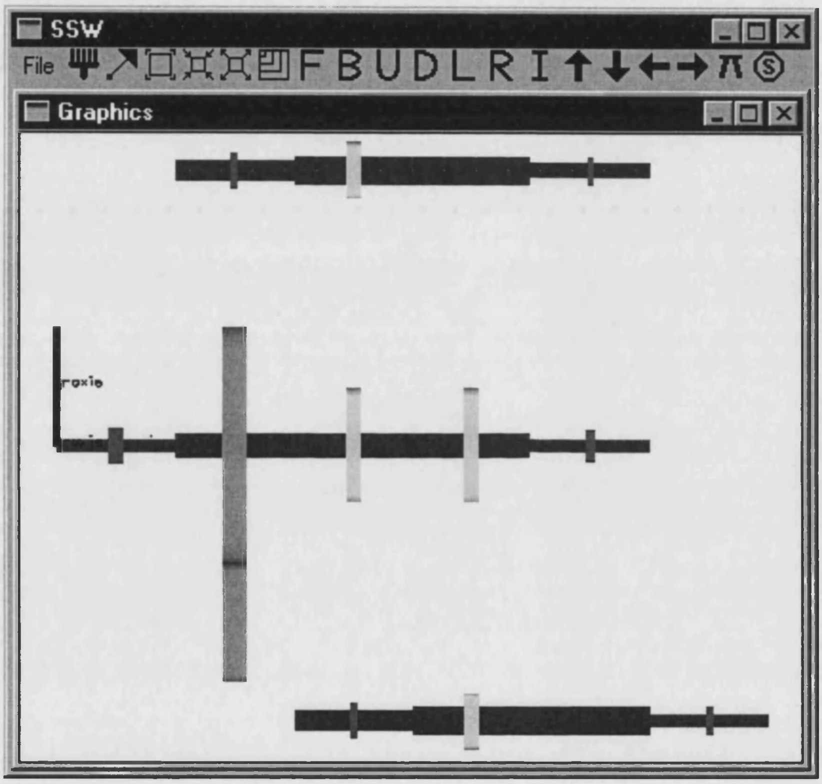


Figure 11.18 – Case study 3 system geometry for the current transmission

System Information

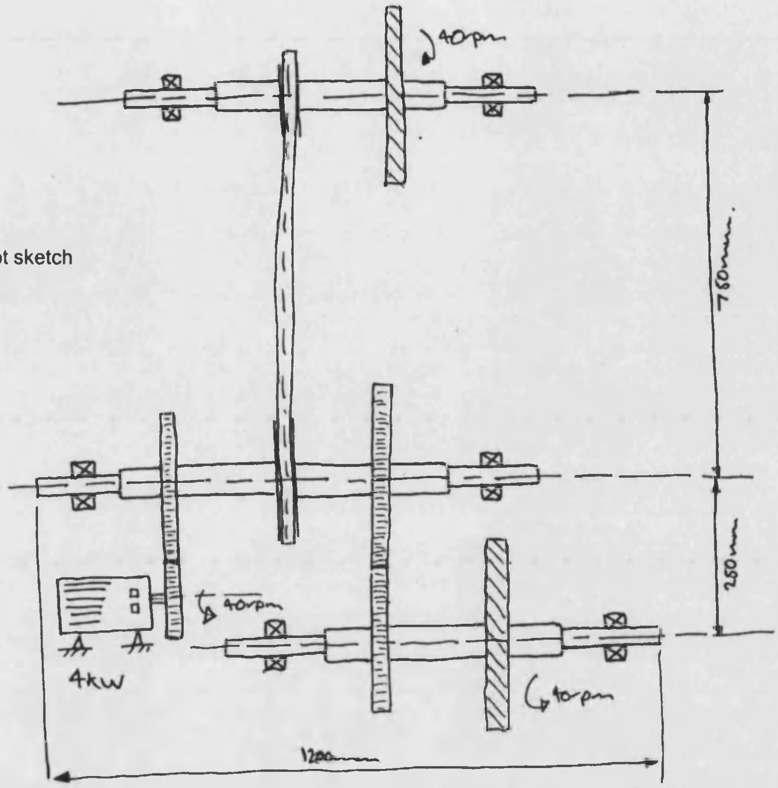
Print

ASSEMBLY PROPERTIES

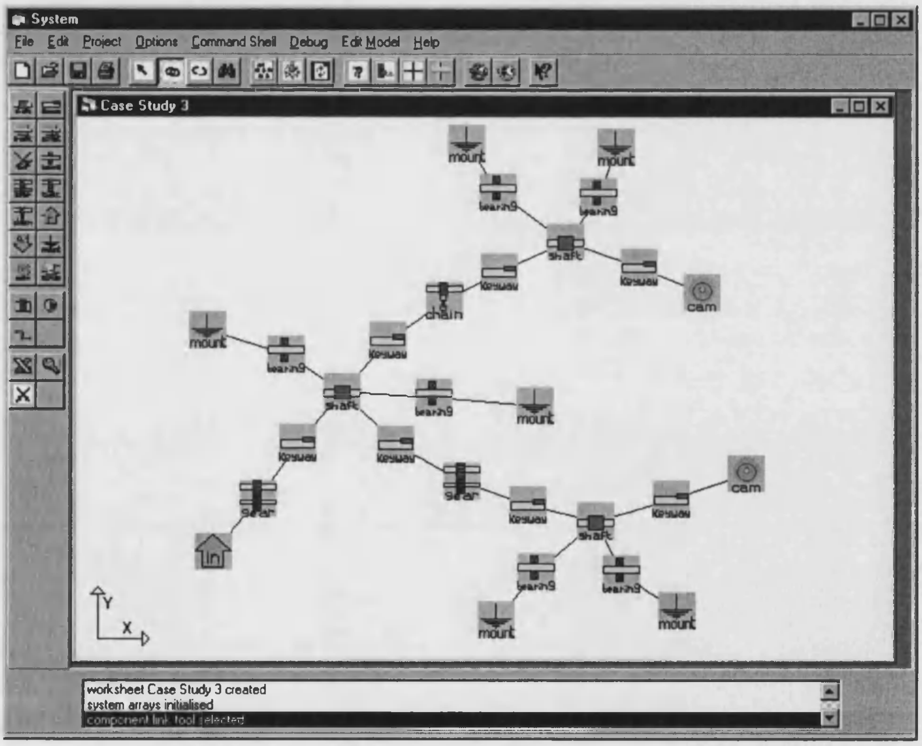
Assembly Mass:	36.00
Assembly Cost:	757.00
Assembly x-dimension:	1200
Assembly y-dimension:	1026
Assembly z-dimension:	200

Index	Component	Mass	Cost
4	Shaft	7.9044	44.9192
5	Key	0.0143	5.00
19	Chain	3.1657	123.9195
6	Key	0.0075	7.00
7	Key	0.0143	5.00
20	Chain	3.1657	123.9195
8	Key	0.0075	7.00
9	Shaft	7.8928	43.691
10	Shaft	7.8928	43.691
17	Bearing	0.16	40.00
21	Bearing	0.35	35.00
3	Key	0.0571	7.00
2	Gear	6.1363	111.8193
25	Key	0.015	5.00
18	Bearing	0.084	41.00
22	Bearing	0.16	26.00
23	Bearing	0.16	40.00
24	Bearing	0.084	41.00
26	Key	0.015	5.00
13	Mount	0.00	0.00
11	Mount	0.00	0.00
1	Input	0.00	0.00
27	Cam	0.00	0.00
14	Mount	0.00	0.00
12	Mount	0.00	0.00
15	Mount	0.00	0.00
16	Mount	0.00	0.00
28	Cam	0.00	0.00

Figure 11.19 – Case study 3 system attributes and component information for the current transmission



Part (a) Concept sketch



Part (b) Concept schematic constructed in the modelling environment

Figure 11.20 – Case study 3 revised transmission for the overwrapper

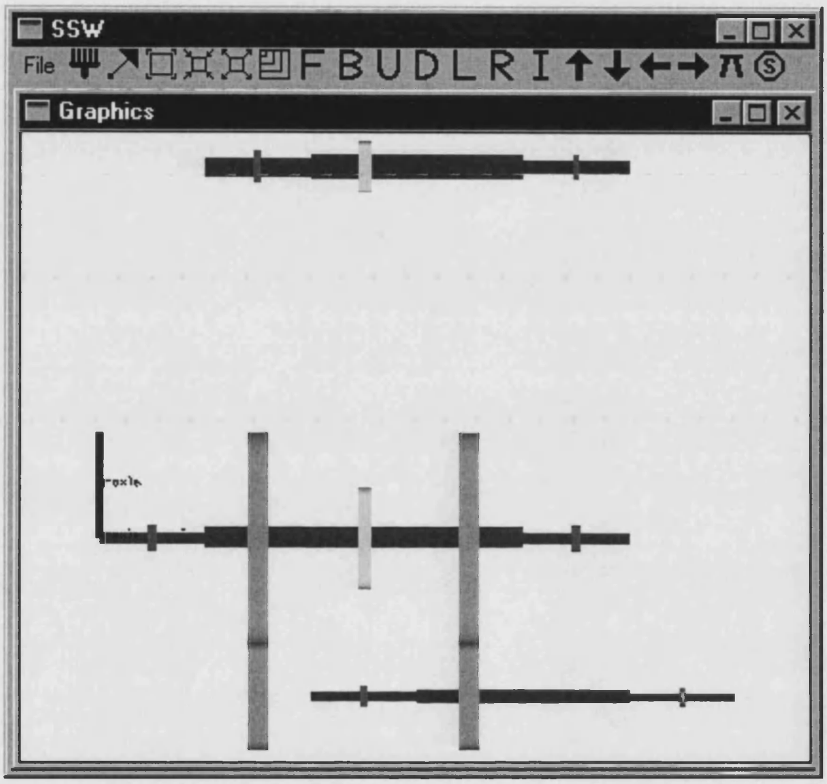


Figure 11.21 – Case study 3 system geometry for the revised transmission

System Information

Print

ASSEMBLY PROPERTIES

Assembly Mass:	33.00
Assembly Cost:	681.00
Assembly x-dimension:	1200
Assembly y-dimension:	1078
Assembly z-dimension:	200

Index	Component	Mass	Cost
4	Shaft	6.6925	39.614
5	Key	0.015	5.00
27	Chain	3.6328	126.8337
6	Key	0.0075	7.00
8	Key	0.015	5.00
28	Gear	6.1363	111.8193
9	Key	0.015	24.00
7	Shaft	7.8928	43.691
10	Shaft	2.1778	16.4928
19	Bearing	0.16	40.00
21	Bearing	0.092	24.00
17	Bearing	0.14	25.00
3	Key	0.03	5.00
2	Gear	6.1363	111.8193
23	Key	0.015	5.00
20	Bearing	0.084	41.00
24	Key	0.0144	5.00
22	Bearing	0.047	19.00
18	Bearing	0.14	25.00
11	Mount	0.00	0.00
13	Mount	0.00	0.00
15	Mount	0.00	0.00
1	Input	0.00	0.00
25	Cam	0.00	0.00
12	Mount	0.00	0.00
26	Cam	0.00	0.00
14	Mount	0.00	0.00
16	Mount	0.00	0.00

Figure 11.22 – Case study 3 system attributes and component information for the revised transmission

	Existing transmission	Revised transmission
No. of components	28	28
System mass (kg)	36	33
Cost (£)	757	681
Leading dimension x (mm)	1200	1200
Leading dimension y (mm)	1026	1078
Leading dimension z (mm)	200	200
Shaft mass (kg)	23.68	16.76

Figure 11.23 – System attributes for the existing transmission and the revised transmission

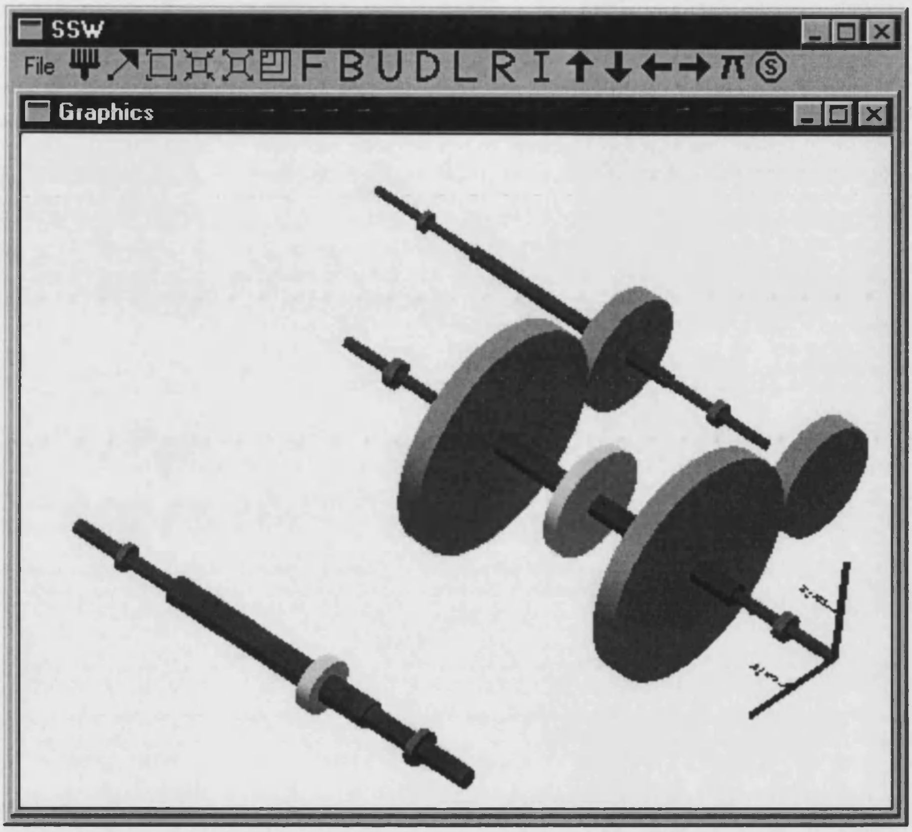


Figure 11.24 – Isometric view of system geometry

Chapter 12

Conclusions

This body of research deals with the development of a modelling approach to support the designer during the transformation of a concept to an embodied solution. This modelling approach provides for the representation of machine systems for the embodiment with a set of real components, and in particular, standard selected and standard designed components. These are either selected from a third party catalogue or designed through standard procedures, and may therefore be produced or procured exactly as specified.

In order to consider real components it is necessary to consider their associated forms of electronic representation within the overall systems approach. The initial focus of the research was to develop the issues and the requirements associated with the development of an integrated modelling approach for machine systems. For the purpose of this work, an integrated modelling approach is one which represents a system by combining and manipulating individual representations for each component. The various issues were developed as the work progresses and are addressed through the creation of appropriate strategies for system representation, handling interactions, system resolution and data arbitration. These strategies are not meant to be optimal implementations for the particular problem, rather they are identified from other work and engineering disciplines, and sufficiently developed to demonstrate the feasibility of the new modelling approach.

It is argued that standard components are a very important factor in the creation of high quality low-cost design solutions and that their utilisation is likely to rise. Consequently, the development of methods which deal with the incorporation of standard components in systems is a particularly important area. A review of research and emerging technologies in engineering design has identified that there are many tools that deal with the electronic selection and design of individual mechanical components. Providing many benefits to both the designer and the supplier, albeit only for the design and selection of an individual component.

For the design of systems of standard components, the designer must consider a large number of performance and geometric attributes at both a component level and a system level. The majority of current systems modelling approaches tend only to consider geometry, although there are some emerging knowledge based engineering tools that do consider limited performance attributes. However, these relationships must be manually configured and programmed by the designer or a programmer. Approaches that consider both performance and geometry generally fall into the category of simulation. This class of modelling tool is widely used in other engineering domains and has been successfully used as 'design through simulation' tools in the fields of electronic circuit design and the design of fluid power systems.

In mechanical engineering, and for power transmission systems involving linear and rotational motion, the development of similar approaches is complicated by a number of issues. These include the vast range and diversity of mechanical components, their equally varied forms of electronic representation and the complex structure of mechanical systems. To overcome this, current computer based tools have had to limit the arrangement and size of the system in order to represent the overall performance of the system. This predefined arrangement is necessary in order to capture the relative complexity of the structure and the constraints between system elements. In addition to this, the discrete nature of the available sizes for standard selected components also frustrates the development of a modelling approach which considers real components. Some attempts have been made to include standard selected components in modelling tools but are severely limited. This is because only continuous or abstracted models of component representations are used and as a consequence only feasible or approximate solutions are determined. Real components must be selected later in the process.

To address these outstanding research issues three hypotheses are proposed.

Hypothesis 1

The electronic representations for standard engineering components can be manipulated in such a manner so as to enable the performance of mechanical systems to be represented.

Hypothesis 2

This approach can be implemented in a computer based support tool to enable the representation of topology and performance for conceptual systems of standard components.

Hypothesis 3

The approach can be extended to enable the configuration, embodiment and optimisation of engineering systems from standard components.

These hypothesis are dealt with in the next three sections.

12.1 Representing the performance of engineering systems from third party electronic representations

A review of existing support tools, their application over the design process, and the requirements of the designer during the embodiment of systems with standard components, has been used to identify and develop the requirements for an integrated modelling environment. These include; representing the performance of the system using component based models, integrating third party or external electronic representations and the process of determining a system that is physically realisable.

A review of modelling approaches and representations for engineering systems and mechanical components has been undertaken. The suitability of techniques such as bond graphs and STEP were evaluated and it was concluded that in their current state of development they lack the capability to represent the necessary level of performance data and geometric data for the identification and selection of standard components within a systems modelling approach. Consequently, a new modelling approach is needed. In the development of this new approach a detailed review of performance modelling and simulation tools in other engineering domains was undertaken. This review has identified three key aspects to the modelling approach that need to be addressed:

- *System representation.* This is the strategy adopted in the modelling approach to represent the type, arrangement and relations between the elements that constitute the system.
- *A protocol for handling interactions.* This encompasses the extents of interactions to be handled within the system and the mechanism for the exchange of data describing these interactions.
- *A procedure for system resolution or analysis.* This is the type and method implemented in the modelling approach to determine or evaluate a particular solution state.

In addition to these aspects, the review also highlighted two support functions; compatibility analysis and data arbitration which are necessary to determine a feasible solution.

For the purpose of system representation, a strategy that represents the connectivity within a mechanical system has been developed. As well as capturing the relative arrangement of elements in the system, this strategy also classifies elements as either *unitary*, *binary* and *principal* elements. These classes are defined in the Definitions section. The incorporation of this classification enables a distinction to be made between the system inputs and outputs (unitary elements) and core components (principal elements), which possess more than two connections.

Binary components possess two connections and occur in sequences of coupled components that convey the system inputs or outputs and link core components. This approach facilitates the generation of a resolution procedure which ensures that all necessary selection data is available prior to the resolution of individual component representations. Firstly, the system inputs and all related components that convey the inputs are resolved. Following which, principal elements are resolved and the outputs determined, the components that convey these outputs are then resolved. This latter phase of the resolution process is repeated for all principal elements.

In order to represent and handle the necessary interactions within a mechanical system, a three-tier classification has been established for component attributes; *global*, *local* and *intrinsic*. These classes relate the dependency of the attribute on the system, constraints imposed by connected components and other intrinsic attributes of the component respectively. Global attributes are dependent upon system level data and intrinsic attributes on component level data, whilst local attributes are dependent on the attributes of other connected components. Because of this, data describing these dependencies is required in order that the components may be effectively designed. The large number of component attributes prevents the explicit exchange of all, or even part, of these local attributes. Consequently, a strategy that exchanges a set of fundamental design parameters has been developed. This range of design parameters was determined through evaluation of a large number of local attributes for mechanical components. These parameters allow the formulation of the full range of local attributes necessary for the design and selection of standard components.

12.2 Implementation of the modelling approach as a computer based support tool

To facilitate the implementation of the modelling approach in a software environment the local design parameters, global attributes, local attributes and intrinsic attributes of components are all communicated within a blackboard structure using an object-oriented approach. In order to develop a new modelling environment that incorporates third party electronic representations, two approaches were considered; creating abstracted models of the representations and interfacing electronic representations with the modelling environment. The first approach has been adopted in previous work and has serious limitations and disadvantages. As a consequence, a strategy for interfacing full electronic representations with the modeller has been developed. Four key features that are necessary for interfacing these representations with the modelling environment were identified.

1 The ability to control or activate the electronic representation.

2 The ability to interrogate and invoke commands within the electronic representation.

3 The ability to exchange information between the modeller and the electronic representation.

4 The ability to construct information and respond meaningfully to information exchanged.

The first three of these features can be addressed in part by enabling interprocess communication between the software environments at runtime. For the purpose of this work, and in particular the embodiment of mechanical systems with standard selected and standard designed components three generic classes of electronic representations have been identified; databases, those built within proprietary software environments and standalone numerical codes. To enable the interfacing of these classes, the modeller incorporates standardised interfaces for a database, a spreadsheet, an ActiveX component and an ASCII data file.

The fourth requirement, for meaningful exchange of information between the modeller and the electronic representation, can only be met by the inclusion of translators for each software entity. These translators compose meaningful queries for the appropriate representation and interpret the results for the incorporation of the data within the modelling environment.

12.3 Supporting configuration, embodiment and optimisation in an integrated modelling environment

The third hypothesis of this work is addressed in two parts. The first deals with assisting the embodiment of a mechanical system and the second with the issues associated with the strategic and optimal design of systems of standard components. In order to assist the embodiment of a machine system model, this research has identified two necessary support functions:

- *Data arbitration* to ensure the embodied solution is free from conflicts and ambiguities.
- *Compatibility analysis* to ensure that mechanically coupled elements are appropriate and complementary.

For the purposes of data arbitration, the factors that cause conflicts were identified and a strategy has been developed to allow their automatic resolution by virtual agents. This research has also identified the need to represent and satisfy secondary constraints. These occur between related components, which may not be physically connected, and include attributes such as material or lubrication type. These constraints are handled at a system level by specifying constraints between the attributes of various components. These constraints are evaluated after the system has been resolved, and the designer is notified of any constraints that are violated.

For compatibility analysis, the various issues regarding the selection of mechanically coupled components have been identified and a distinction is made between achievable[1] designs and preferred[2] designs. To evaluate the compatibility of mechanical components three distinct activities have been proposed:

- *Connectivity analysis* ensures that the geometric interfaces are matched and that energy interfaces are compatible.
- *Performance matching* ensures that the magnitudes of energy transfer are acceptable output and input levels for coupled components.
- *Complementary assessment* provides for the qualitative considerations which the designer must undertake.

A strategy which provides for the distinction of compatible and incompatible elements as well as preferred components has been developed. This involves the incorporation of a knowledge base in the modelling environment. This explicitly represents the relationship between every possible component pairing. The combination of data arbitration and compatibility analysis ensures that 'achievable' designs can be automatically configured and that 'preferred' components are used where possible.

The creation of the integrated modelling environment provides a tool for the comparison and evaluation of design alternatives which bounds a real solution space, and as such affords a platform for optimisation. For the strategic design or optimisation of any system four important factors have been identified; namely the satisfaction of performance requirements, the evaluation of cost, the evaluation of mass and the determination of spatial occupancy. The modelling approach deals with the performance requirements of the system; whilst the inclusion of the real component data enables the generation of spatial occupancy, which is represented by the leading dimensions in the x, y and z planes; and mass, which is obtained by summing the attributes of individual elements.

---

[1] Achievable designs comprise a set of 'real' components and are fundamentally based on existing technology or principles and contain a high proportion of standard components. Discussed in chapter 6.

[2] Preferred components are those elements that are perceived to deliver better performance. Reasons for this may include improved loading capabilities, reliability, reduced costs or favoured suppliers. Discussed in chapter 6.

However, the consideration of cost is frustrated by the fact that the majority of component representations do not include cost data. To overcome this, cost forecasting techniques for the purposes of selection design have been developed for each of the three classes of engineering component; standard selected, standard designed and bespoke designed. The requirement for cost modelling at the early stages of design, and in particular the transformation from concept to an embodied solution, is primarily to describe the relative change in magnitude of costs within a component range and to provide a comparison of cost between component types rather than to establish absolute values. Three classes of cost model have been developed:

1 A *specific cost model* quantifies the cost behaviour of a particular component type and range.

2 A *universal cost model* is used in situations where it is necessary to represent the costing structure of either a type of mechanical component or a family of like products.

3 A *configuration cost model* aims to capture the costs incurred through changes in the parameters or attributes of a bespoke designed component.

These approaches are demonstrated and validated through the application of the techniques to a number of mechanical components and a case study.

The integrated modelling environment provides a 'real' solution space and with the availability of cost, mass and spatial occupancy provides a platform for multi-objective optimisation. However, a number of issues must be addressed. These issues have been developed in this work, and include problem formulation, dealing with discrete elements and system resolution. Furthermore, the requirements for the integration and control of a third party optimiser with the modelling environment are discussed. However, the reliable treatment of the optimisation problem and the incorporation of the strategy into the modelling environment is beyond the scope of this work.

12.4 Validation of the approach

The hypotheses originally set out for this work are validated in two parts. Hypothesis one and two are demonstrated through the implementation of the integrated modelling approach within the software environment, described in chapter 8. This demonstrates the feasibility of the modelling approach and that a systems approach can be achieved by the combination and manipulation of various individual electronic representations. Hypothesis two and the third hypothesis are demonstrated through the application of the modelling tool to a number of industrial case studies, described in chapter 11. This demonstrates both the functionality and novelty of the modelling approach. In particular, it shows how the approach can assist the embodiment of conceptual systems with standard components and facilitate the strategic and optimal design of systems. The case studies illustrate and validate the various capabilities of the new modelling approach, and include the ability to:

- Handle and automatically embody a design configuration for different performance requirements.
- Automatically embody different configurations or design concepts for the same or similar performance requirements.
- Assist the embodiment of systems to meet essential physical/spatial requirements.
- Represent and handle large complex systems comprising multiple assemblies or subassemblies. The first case study comprises a single assembly with eleven elements, the second consists of two assemblies and nineteen components, whilst the third case study comprises three assemblies and twenty eight components.
- Evaluate the effects upon the overall system which are brought about by the introduction of a changed part or redesigned/reconfigured assembly.

The successful application of the software tool to these cases demonstrates the feasibility of this new modelling approach and illustrates its capabilities which all validate the original hypotheses.

12.5 Concluding remarks

The work in this thesis deals with the development of an integrated modelling environment for the embodiment of machine systems with standard components. The work has demonstrated the need to support the building and embodiment of mechanical systems from standard components. In order to achieve this, a new modelling approach has been created which considers the performance and geometry at both a system level and a component level. Furthermore, the approach provides a flexible and unrestricted representation of a system both in terms of its size and structure. For the representation of individual components, the wealth of electronic representations available are reviewed and their advantages as design and selection tools discussed. Consequently, the benefits of interfacing these representations with a systems modelling approach are many. To achieve this, the modelling approach developed in this work, represents the system as a whole, whilst maintaining the integrity of the component based representations. This is enabled through the development of standard procedures for the integration of the various classes of electronic representation with the modelling tool. Because existing (third party) component representations are integrated in the modelling tool, the approach ensures that 'real' components are considered, which is essential for effective system design.

The capabilities and novelty of the new modelling approach are demonstrated through the application of the modelling environment to three case studies. Of particular interest, is the ability of the approach to support the designer in configuring, embodying and analysing design solutions. During these tasks the designer must consider a large number of components and consider an extensive range of attributes. For example, the third case study comprises twenty eight components, each of which possesses between ten and twenty four different attributes. If a system of this size were embodied manually by the designer, the process could take many hours, involving a number of analytically intensive and error prone tasks. However, the modelling approach is capable of handling and analysing such a system in only a fraction of the time. Thereby reducing the time taken to embody a design solution and allowing the designer to evaluate many more design alternatives. This enables the development of a more refined design and ultimately more fully informed decisions to be taken at an early stage in the design process.

12.6 Directions for future work

One of the main objectives of this research was to demonstrate the feasibility of an integrated modelling approach for the building and embodiment of mechanical systems with standard components. The work has developed the requirements for a system modelling approach and identifies the necessary software features and support functions to enable the approach. In so doing, many opportunities for future work are identified which build upon and extend the issues identified and addressed within this body of research. Some of these issues are highlighted below but are by no means exhaustive:

- This work has reviewed the various standards for representing engineering components. The majority of this work focuses on geometry. For the purposes of selection design a wealth of performance data must also be represented. The investigation and development of standards for representing this performance data is essential for the integration of the vast number of electronic selection procedures. This common representation would enable meaningful exchange of data between modellers and component based design and selection software.

- An area identified through this work is the provision of methods for accessing the data within component representations. This separation of presentation, functionality and data can be achieved by using an independent data structure such as XML. Furthermore, XML is similar in its approach to EXPRESS the transport language for STEP. It may therefore be possible to develop a STEP based implementation for data describing the performance attributes of standard components as well as the geometric attributes, which can be accessed remotely by other applications and used during component selection. This would enable the interrogation of multiple representations for, say, a bearing or a gear in order to obtain an optimal component match.

- In order to determine a solution state for any complex system, methods which arbitrate data and resolve conflicts are essential. A more detailed investigation of the considerations and various objectives/approaches of the designer during the embodiment and decision making phase of the design process is necessary in order to develop appropriate strategies for the negotiation and resolution of conflicts during the automatic embodiment of systems with standard components.

- A novel extension of the overall methodology is the provision for the automatic synthesis of standard system configurations from functional descriptions provided by the designer. This may either combine assemblies or alter existing design configurations.

- The cost of the system is a very important consideration. Most of the emerging electronic representations do not include costing information. Consequently, techniques that represent the cost of components must be incorporated. The current implementation only provides costing information for each component. The cost of procurement and system assembly are not considered but are important considerations in the selection of design alternatives. Approaches which deal with the resources required for design, assembly and manufacture could be incorporated into the approach.

- The integrated modelling environment provides a 'real' solution space and with the availability of cost, mass and spatial occupancy provides a platform for multi-objective optimisation. This work has developed the issues associated with the application of optimisation procedures to the modelling environment. However, the reliable treatment and incorporation of optimisation algorithms afford a number of outstanding research issues.

- Many electronic selection procedures are utilising the Internet as a medium. This has the advantage of being able to convey up-to-date and accurate information. This can include order times, current stock, costing information and even discounts. The investigation of methods, which interface these web-based tools within the modelling approach, is a novel area for future work.

- The implementation of the approach in a software environment would also demand considerable development. This includes refinements of the underlying software architecture necessary for an optimal implementation and issues over usability and human computer interaction.

Related publications

Conference Proceedings

1. Hicks, B.J. & Culley, S.J., (1999) "A flexible representation of engineering assemblies for an integrated modelling environment", *Proceedings of ICED99*, International Conference on Engineering Design, Munich, August 1999, pp. 1113-1116, ISBN 3-922979-53-X.

2. Allen, R.D., Hicks, B.J. & Culley, S.J., (2000) "Integrating electronic information for the design of mechanical systems: The designers' perspective", World Multiconference on Systemics, Cybernetics and Informatics, *SCI 2000 and ISAS 2000*, July 23, 2000 pp. 266-271.

3. Hicks, B.J., Culley, S.J. & Mullineux, G., (2001) "Integrating engineering component representations for the design and synthesis of mechanical systems", *ICED 01,* International Conference in Engineering Design, August 2001, pp. 437-444, ISBN 1-86058-355-5.

Journal Publications

4. Hicks, B.J. & Culley, S.J., (2000) "A protocol for communication in a component based modelling infrastructure", Institution of Mechanical Engineers Journal of Engineering Manufacture, Part B, Vol. 215, No. B4, ISSN 0954-4054, pp. 453-464.

5. Hicks, B.J. & Culley, S.J., (2002) "An Integrated Modelling Environment for the Embodiment of Mechanical Systems", Journal of Computer-Aided Design, Vol. 34, No. 6, pp. 435-451.

6. Hicks, B.J., Culley, S.J., Allen, R.D., & Mullineux, G., (accepted 2001) "A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design", International Journal of Information Management.

7. Hicks, B.J., Culley, S.J. & Mullineux, G., (accepted 2001) "Cost estimation for standard components and systems in the early phases of the design process", Journal of Engineering Design.

8. Hicks, B.J. & Culley, S. J., (accepted 2001) "Compatibility issues for mechanical system modelling with standard components", Institution of Mechanical Engineers Journal of Engineering Manufacture.

9. Allen, R.D., Hicks, B.J., & Culley, S.J., (submitted 2001) "Integrating electronic information for mechanical systems design", Journal of Technology Management.

10. Hicks, B.J., Culley, S.J. & Mullineux, G., (submitted 2002) "The representation of engineering systems for their embodiment with standard components", ASME, Journal of Mechanical Design.

References

Adeli, H., 1994, "*Advances in Design Optimisation*", Chapman & Hall, London, UK, ISBN 0-412-53730-2.

Aldrich, R.L., 1995, "Costing a bill of material/route of a thermoformed part using integrated MRPII software", *Technical Paper – Society of Manufacturing Engineers*, 1995.

Allen, R.D., Hicks, B.J. & Culley, S.J., 2001, "Integrating electronic information for mechanical systems design", *Journal of Technology Management* (submitted).

Allen, R.D., Hicks, B.J. & Culley, S.J., 2000, "Integrating Electronic Information for the design of Mechanical Systems: The Designers Perspective", 4th *World Multi-Conference on Systematics, Cybernetics and Informatics*, Vol. 2, pp. 266-271.

AMESim, 2000, "Fluid system and power train simulation", http://www.amesim.com/amesim.htm.

Anderson, J.D., 1995, "*Computational Fluid Dynamics: The Basics with Applications*", McGraw Hill, Inc., ISBN 0-07001-685-2.

Anon, 1997, ABAQUS User Manual, Hibbitt, Karlsson & Sorenson, Inc. USA.

Ashby, M.F., 1999, "*Materials Selection in Mechanical Design*", Second Edition, Butterworth Heinemann, Oxford, UK, ISBN 0 7506 4357 9.

Autodesk Inc, 1997, "AutoCAD Release 14 User's Guide".

Bahler, D., Bowen, J., O'Grady, P. & Young, R.E., 1990, "Constraint networks for concurrent engineering", *Issue in design/manufacturing integration*, DE-Vol.29, pp.59-64.

Bancroft, N.H., Seip, H. & Sprengel, A., 1998, "*Implementing SAP R/3*", Manning Publications Co., Greenwich, USA, ISB 0-13-889213-X.

Baya, V. & Leifer, L., 1995, "Understanding Design Information Handling Behaviour Using Time and Information Measure", *ASME Design Engineering*, Vol. 83, No. 2, pp. 555-562.

Blundell, A., 1982, "*Bond Graphs for Modelling Engineering Systems*", UK: Ellis Horwood Limited, ISBN 0-85-312510-4.

Boothroyd Dewhurst Inc, 2001, "DFMA Design for Manufacture and Assembly Homepage", http://www.dfma.com.

Bordegoni, M. & Cugini, U., 1997, "Feature-based Assembly Design: Concepts and Design Environment", *Proceedings of Design Engineering Technical Conference, ASME*, California, pp. 1-8.

Boston, O., Culley, S.J. & McMahon, C.A., 1998, "Life-Cycle Management of Supplier Literature: The Pertinent Issues", *Journal of Product Innovation Management.*

Boston, O., 1999, "*Technical Liaisons in Engineering Design Understanding By Modelling*", Ph.D. Thesis, University of Bath, UK.

Boston, O. P., Culley, S.J. & McMahon, C., 1999, "Suppliers In New Product Development: Their Information and Integration", *Journal of Engineering Design*, Vol 10. No 1, pp. 59-75.

Bowler, C., Medland, A.J. and Mullineux, G., 1999, "Communication of parametric data between design processes", *Proceedings International Conference on Engineering Design*, ICED 99, Lindemann, U., Birkhofer, H., Meerkamm, H. and Vajna, S., eds., Munich, August 1999, 1737-1740, ISBN 3-922979-53-X.

Brady, D. & Juster, N.P., 1997, "The Creation of Assembly Geometry from Functional Definitions", *Proceedings of ASME Design Engineering Technical Conferences*, Sacremento California.

Breeveld, P., 1982, "Proposition for an unambiguous Vector Bond graph notation", *Journal of Dynamic Systems, Measurement and Control*, Vol. 104, No. 3, pp. 267-270.

Brimson, J.A., 1998, "Feature costing: Beyond ABC", *Journal of Cost Management*, Lanuary-February 1998, pp. 6-12.

Brinkop, A., Laudwein, N. & Maasen, R., 1995, "Routine Design for Mechanical Engineering", *Artificial Intelligence Magazine*, Spring 1995, AAAI. 0738-4602-1995, pp. 74-84.

Brown, D. C., 1999, "Some Thoughts on Configuration Processes", http://www.cs.wpi.edu/~dcb/config.htm.

Brown, F.T., 1991,"Hamiltonian and Lagrangian Bond Graphs", *J.Franklin Institute*, Vol. 328, No. 5/6, pp. 809-831.

BS 7000, 1999, "Design management systems. Guide to managing innovation", British Standards 7000 Part 1.

BS 436-1, 1967, Spur and Helical Gear Design, Basic rack form, pitches and accuracy (diametral pitch series), 1967.

BS 3170, 1972, Specification for flexible couplings for power transmission.

Carter, D.E. & Baker, B.S., 1992, "*CE: Concurrent Engineering The Product Development Environment for the 1990's*", Addison Wesley, ASIN: 0201563495.

Cartmell, M.P., Fothergill, A.P. & Foster, J.K.W., 1993, "Theoretical Foundations for a Design Brief Expansion System", *Journal of Engineering Design*, Vol. 4, No. 3, pp. 221-247.

Claypole, J., 1993, "Pondering on Parametrics", Professional Engineering, March 1993, pp. 22-23.

Clews, G. & Leonard, R., 1985, "*Technology and Production*", Philip Allan Publishers Limited, Oxford, England, ISBN 0-86003-527-1.

Coleman, J.M., McGuff, P.C. & Koederitz, W.L., 1996, "PC compatibility – is there such a thing anymore?", *Proceedings Petroleum Computer Conference*, pp.129-134.

Collins Concise Dictionary, 1990, William Collins Sons & Co. Ltd., Glasgow, UK, ISBN 0-19-861200-1.

Cornell, G., 1999, "*Visual Basic From The Ground* Up", Osborne/McGraw-Hill, ISBN 0-07-882349-8.

Counsell, J., Porter, I., Dawson, D. & Duffy, M., 1999, "Schemebuilder: computer aided knowledge based design of mechatronic systems", *Assembly Automation*, Vol 19, No. 2, pp. 129-138.

Court, A.W., 1995, "*Modelling and Classification of information for Engineering Designers*", Ph.D. Thesis, University of Bath.

Culley, S.J., 1999, "*Future Issues For Design Research*", FIDR workshop, University of Bath, Bath, UK, ISBN 1-85790-061-8.

Culley, S.J. & Allen, R.D., 1999, "Informal information – definitions and examples with reference to the electronic catalogue", *Proceedings International Conference on Engineering Design*, ICED 99, pp. 1961-1964.

Culley, S.J. & Theobold, G., 1997, "Dealing with standard components for knowledge intensive CAD", *Knowledge intensive CAD,* Vol.2, Proc. of the IFIP TC5 WG5.2 International conference on knowledge intensive CAD, (Ed. Mantyla, M. Finger, S. and Tomiyama, T.) Chapman and Hall, 1997.

Culley, S.J. & Webber, S.J., 1992, "Implementation requirements for electronic standard component catalogues", *Proceedings Institution of Mechanical Engineers*, Journal of Engineering Manufacture: Part B. Vol. 206, pp. 253-260.

Culley, S.J. & Vogwell, J., 1990, "The Role of Feedback in Computer Aided Component Selection Systems", *The Canadian Society for Mechanical Engineering,* CSME Engineering Forum 1990, pp. 289-293.

Culley, S.J., Vogwell, J. & Corderoy, M.P., 1990, "Assembly considerations with standard components from engineering databases", *Proceedings International Conference on Engineering Design*, ICED 90, Dubrovnik, August 1990.

Curtis, G., 1991, "Business Information Systems: Analysis, Design and Practice", Addison-Wesley Publishing Company, UK, ISBN 0-20117-523-1.

Davis, B., 1998, "Review of IM Phenomenon: Mr Altschullers' big idea", *The Engineer*, May 1998.

EAI, 2000, "Engineering Animation Inc, Jack® virtual mannequin for ergonomics and human factors", http://www.eai.com.

EDS Inc., 2000, "Introducing I-DEAS", http://www.sdrc.com/ideas/index.htm.

Ellram, L.S., 1990, "The Supplier Selection Decision in Strategic Partnerships" *Journal of Purchasing and Materials Management*, Vol. 32, No. 18, pp.8-14.

Engineous Software Inc., 2001, "iSight Exploration Engine", http://engineous.com/overview.html.

Evbuomwan, N.F.O., Sivaloganathan, S. & Jebb, A., 1996, "A survey of design philosophies, models, methods and systems", *Proceedings Institution of Mechanical Engineers*, Part B: Journal of Engineering Manufacture, Vol. 210, pp. 237-255.

Fagan, M. J., 1992, "Finite Element Analysis: Theory and Practice", Prentice Hall, London, UK, ISBN 0-582-02247-9.

Finger, S. & Dixon, J.R., 1989, "A Review of Research in Mechanical Engineering Design - Part I: Descriptive, Prescriptive and Computer-Based Models of Design Processes", *Research in Engineering Design*, Springer-Verlag, New York Inc., pp. 51-67.

Finger, S. & Dixon, J.R., 1989, "A Review of Research in Mechanical Engineering Design - Part II: Representations, Analysis and Design for Life Cycle", *Research in Engineering Design*, Springer-Verlag, New York Inc., pp. 121-137.

Flood, R.L. & Carson, E.R., 1988, "Dealing with Complexity: An Introduction to Theory and Application of Systems Science", ISBN 0 306 42715, Plenum Press, New York, UK.

Foussier, P., 1998, "Can statistics be used for cost estimating purposes?", *The Cost Engineer*, Vol. 36, No. 6, pp. 8-11.

FPT Ltd, 1999, Fenner Power Transmission Drive Design and Maintenance Manual.

French, M.J., 1990, "Function Costing: A Potential Aid to Designers", *Journal of Engineering Design*, Vol. 1, No. 1.

French, M.J., 1988, "Invention and Evolution", Cambridge University Press, UK, ISBN 0-52131-492-5.

Gawthrop, P. & Smith, L., 1996, "Metamodelling: Bond graphs and dynamic systems", *International series in Systems Modelling and Control Engineering*, Prentice Hall International (UK) Limited, ISBN 0-13489-824-9.

Gould, H. & Tobochnik, J., 1998, "*An Introduction to Computer Simulation Methods: Applications to Physical Systems*", Addison-Wesley, ISBN 0-20116-504-X.

Hajare, A.D., 1998, "Parametric Costing – steel wire mill", *Proceedings of the Annual Convention of the Wire Association International*, pp. 172-178.

Hajela, P., 1999, "Nongradient Methods in Multidisciplinary Design Optimization – Status and Potential", *Journal of Aircraft Design*, Vol. 36, No. 1, January-February 1999, pp. 255-265.

Hales, C., 1993, "*Managing Engineering Design*", Longman Scientific & Technical, ISBN 0-582-03933-9.

Harmer, Q.J., Weaver, P.M. & Wallace, K.M., 1998, "Design-led Component Selection", *Computer-Aided Design*, Vol. 30, No. 5, pp. 391-405.

Hicks, B.J. & Culley, S.J., 2001a, "Compatibility issues for mechanical system modelling with standard components", *Proceedings Institution of Mechanical Engineers*, Part B: Journal of Engineering Manufacture (accepted for publication).

Hicks, B.J. & Culley, S.J., 2001b, "A protocol for communication in a component based modelling infrastructure", *Proceedings Institution of Mechanical Engineers*, Part B: Journal of Engineering Manufacture, Vol. 215, No. B4, pp. 453-464.

Hicks, B.J., Bowler, C., Medland, A.J, & Mullineux, G., 2000, "A redesign methodology for analysing and improving the performance capability of packaging machinery", *International Journal of Industrial Engineering*, (submitted).

Holt, K., 1991, "The impact of technology strategy on the engineering design process", *Design Studies*, Vol. 12, No. 2, April 1991.

HPC Gears, 1997, Catalogue C15, HPC Gears Ltd, Chesterfield, England.

Hubka, V., 1982, "Principles of Engineering Design", Butterworth Scientific, ISBN 0-40801-105-X.

Hubka, V., 1988, "Practical Studies in Systematic Design", Butterworth & Co, ISBN 0-40801-420-2.

Hundal, M.S. & Langholtz, L.D., 1992, "Conceptual Design by Computer-Aided Creation of Function Structures and the Search for Solutions", *Journal of Engineering Design*, Vol. 3, No. 2, pp. 127-138.

INBIS Group Plc, 2001, "INBIS Constraint Analyser", http://www.inbis.com/software/index.htm.

Johnson, A.L. & Thornton, A.C., 1991, "Towards Real CAD", *Design Studies*, Vol.12, No. 4, pp. 232-236.

Karnopp, D., 1982, "Dynamic forces in Multiport Mechanics; Direct and Lagrangian Formulations", *J.Franklin Institute*, Vol. 314, No. 5, pp. 265-270.

KBS Inc, 2001, "IDEF Family of Methods", http://www.idef.com.

Keown, J., 1994, "Pspice and Circuit Analysis", Macmillan Publishing Company, UK, ISBN 0-02363-526-6.

Keynote, 1998, *Aerospace Transmission Systems – Concurrent Design and Manufacture*, Institution of Mechanical Engineers Seminar Publication, ISBN 1-86058-161-7.

Kielkowski, R., 1994, "Inside SPICE, Overcoming the obstacles of circuit simulation", McGraw-Hill, Inc., ISBN 0-07911-525-X.

Klein, M. & Lu, S.C.Y., 1990, "Conflict Resolution in co-operative design", *Artificial Intelligence in Engineering,* Vol. 4 No.4, pp. 168-180.

Knutton, P., 1993, "Spending Heads in the Right Direction", The 1993 Computers in Manufacturing Survey, *Engineering Computers,* November/December 1993.

Konda, S., Monarch, I., Sargent, P. & Subrahmanian, E., 1992, "Shared Memory in Design", A Unifying Theme for Research and Practice", *Research in Engineering Design*, No. 4, pp. 23-42.

Krick, E.V., 1969, "*An introduction to engineering and engineering design*", Wiley & Sons, UK.

Kusiak, A. & Larson, N., 1995, "Decomposition and representation methods on mechanical design", American Society of Mechanical Engineers, *Journal of Mechanical Design*, June 1995, Vol. 117, pp. 17-24.

Law, D.J., 1993, "Integrated Tools and Technologies to Support Concurrent Engineering", *Effective Technologies for Engineering Success – Making CAD/Cam Pay,* MEP Ltd.

Laudon, K.C. & Laudon, J.P., 1998, "*Management Information Systems: New Approaches to Organisation and Technology*", Prentice Hall, New Jersey US, ISBN 0-13-857723-4.

Leibl, P., Hundal, M. & Hoehne, G., 1999, "Cost Calculation with a Feature-based CAD System using Modules for Calculation, Comparison and Forecast", *Journal of Engineering Design,* Vol. 10, No. 1, pp. 93-102.

Mailoi, C., Penzo, W., Sola, S. & Vitali, F., 1994, "Using a reference model for information system compatibility", *Proceedings Hawaii Conference on Systems Science*, Vol. 3.

Mansour, F.F., 1999, "Perceptions and reality of value engineering", *The Cost Engineer,* November 1999, pp. 17-19.

Mantyla, M. & Gui, J., 1994, "Functional understanding of assembly modelling", Computer-Aided design, Vol 26, No. 6, pp. 435-451.

McMahon, C. & Browne, J., 1993, "*CAD CAM from principles to practice*", Addison-Wesley Publishing Company, Wokingham, England, ISBN 0-210-56502-1.

Medland, A.J., 1990, " Whole in one", *Engineering Design and Technology,* Spring 1990, pp. 16-18.

Medland, A.J. & Burnett, P., 1986, "*CADCAM In Practice*", Kogan Page Ltd., London, ISBN 0 85038 817 1.

Medland, A.J., Mullineux, G., Rentoul, A.H. & Twyman, B.R., 1996, "A constraint based approach to the design and optimisation of mechanism systems", *IUTAM Symposium on Optimization of Mechanical Systems*, Kluwer Academic Publishers, pp. 205-212.

MicroSim Corporation, 1997, MicroSim DesignLab, CD-ROM literature, Irvine, USA.

Microsoft, 2001a, "Comparing OLE & DDE Message Protocols and Libraries", http://www.microsoft.com/support/kb/articles/.

Microsoft, 2001b, "Microsoft supported standards", http://www.microsoft.com/support/kb/standards.

Mileham A.R., Currie G.C., Miles A.W. & Bradford D.T., 1993, "A parametric approach to cost estimating at the conceptual stage of design", *Journal of Engineering Design*, Vol. 4, No. 2, pp 117-125, ISSN 0954-4828.

Mittal, S. & Frayman, F., 1989, "Towards a Generic Model of Configuration Tasks", *Proceedings International Joint Conference on Artificial Intelligence* (IJCAI), Knowledge Representation, pp. 1395-1400.

Moaveni, S., 1999, "Finite Element Analysis: Theory and Application with ANSYS", Prentice Hall, New Jersey, 0-13-785098-0.

Model Engineer, 1993, "Brunell's Blocks: Models of the 1st Mass Production System", August 6th-19th 1993, Vol. 171, No. 3950, pp. 125-154.

Morgan, S., 1988, Chief Transmissions Engineer, Chair of IMECHE Seminar, Afternoon session, *Aerospace Transmission Systems – Concurrent Design and Manufacture.*

MSC Software, 2001, "MSC visual Nastran 4D and Working Model 2D", http://www.workingmodel.com.

NRC, 1991, Executive Summary, "*Improving Engineering Design*", National Research Council, 1991, ISBN 0-309-04478-2.

Pahl, G. & Beitz, W., 1996, Translated by Ken Wallace, "*Engineering Design: A Systematic Approach*", Second Edition, Springer-Verlag, ISBN 3-540-19917-9.

Pahl, G. & Beitz, W., 1984, *"Engineering Design"*, The Design Council, Springer-Verlag, ISBN 0-85072-124-5.

Papalambros, P.Y. & Wilde, D.J., 1988, *"Principles of Optimal Design: Modelling and Computation"*, Cambridge University Press, New York, ISBN 0-521-30674-4.

Parry, J., Bailey, C. & Aldham, C., 2000, "Multiphysics Modelling for Electronic Design", *ITherm 2000*, Seventh Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems.

PDES Inc., 2000, "Results of STEP Testing Worldwide", http://www.pdesinc.scra.org/step_overview.html.

Phadke, M.S., 1989, *"Quality Engineering using Robust Design"*, Prentice-Hall International Editions, ISBN 0-13745-167--9.

Pierra, G., 1994, *"Parametric Product Modelling for STEP and Parts Library"*, Draft proposal. ISO/WD 10303-11:1998, Product Data Representation and Exchange, EXPRESS Language Reference Manual.

PTC Inc., 2001, "Pro/MECHANICA", http://www.ptc.com/products/proe/sim/index.htm.

PTC Inc., 2000, "Pro/ENGINEER Solutions", http://www.ptc.com/products/proe/index.htm.

Pugh, S., 1991, *"Total Design"*, Addison-Wesley Publishing Company, New York, ISBN 0-201-41639-5.

Pye, A., 1991, "Workflow Management Across Computer Systems", *Design Engineering*, September 1991, pp. 43-46.

Qian, L. & Gero, J.S., 1996, "Function-Behaviour-Structure Paths and Their Role in Analogy-Based Design", *Artificial Intelligence in Engineering Design*, Analysis and Manufacture, Vol. 10, pp. 289-312.

Redfield, R.C., 1996, "A Synthesis of Design and Lagrangian Bond Graphs", Proceedings Aamerican Society of Mechanical Engineers, 1996 Dynamics, Systems and Control Division, DSC-Vol. 58, pp. 715–720.

Reinemuth, J., 1993, "Designing with Computer-Based Component Catalogues using Hypermedia", *Proceedings of the International Conference on Engineering Design,* ICED 1993, August 1993, The Hague, pp. 1700-1707.

Richards, T.G., Hughes, E.J. & Tilley, D.G., 1999, "A Multimedia Approach to Fluid Power Systems Design", *proceedings IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, pp. 792-796.

Robinson, C.G., 1991, "Product costing: An essential element for success in bare board manufacture", *Circuit World*, Vol. 17, No. 4, pp. 27-28.

Rosen, D.W., Bras, B.A., Mistree, F. & Goel, A., 1995, "Virtual Prototyping for Product Demanufacture and Service Using a Virtual Design Studio Approach", *Computers in Engineering Conference and the Engineering Database Symposium*, Boston, Massachusettes, pp. 951-958.

Rosenman, M. A. & Gero, J. S. 1997, "Collaborative CAD modelling in multidisciplinary design domains", *in* Maher, M. L., Gero, J. S. and Sudweeks, F. (eds), *Preprints Formal Aspects of Collaborative Computer-Aided Design*, Key Centre of Design Computing, University of Sydney, Sydney, pp. 387-403.

Roztocki, N. & LaScola, K., 1999, "Integrating activity-based costing and economic value added manufacturing", *Engineering Management Journal*, Vol. 11, No. 2, pp. 17-22.

RS Components Limited, 1997, *RS Catalogue Mechanical Components,* Vol. 5, section 36, 1997.

Samuel, A. & Wier, J., 1999, "Introduction to Engineering Design: Modelling, Synthesis and Problem Solving Strategies", ButterWorth-Heinemann, Oxford, UK, ISBN 0-7506-4282-3.

Schemebuilder, 1998, "Schemebuilder: Engineering Knowledge Management", http://www.comp.lancs.ac.uk/edc/schemebuilder.html.

Sen, P. & Yang, J., 1988, "*Multiple Criteria Decision Support in Engineering Design*", Springer-Verlag, New York, USA, ISBN 3-5401-9932-2.

Shigley, J.E. & Mitchell, L.D., 1983, "*Mechanical Engineering Design*", 4th Edition, McGraw-Hill, ISBN 0-07056-881-2.

Silvaco Internatonal, 1998, "SmartSPICE simulation", http://www.silvaco.com/products/smspice/smartwin/smartwin.html.

SKF Limited, 1998, General Catalogue, pp. 329-373.

Sidders, J.A., Tiley, D.G. & Chapple, P.J., 1996, "Thermal-hydraulic performance prediction in fluid power systems", Proceedings of Institution of Mechanical Engineers, Vol. 210, pp. 231-242.

Smith, P.G. & Reinertsen, G., 1995, "*Developing products in half the time*", Van Nostrand Reinhold, ISBN 0-442-02064-3.

Smith. W.A., 1983, "*A guide to CADCAM*", The Institution of Production Engineers, London, UK, ISBN 0-8551-0026-5.

SolidWorks Corporation, 1988, Solidworks 98 Plus User's Guide, USA.

Spiegel, M.R., 1961, "*Theory and problems of statistics*", McGraw-Hill Book Company, pp. 244.

Sreeram, R.T., 2000, "*Resolving conflicts in agent-supported collaborative product development*", Ph.D. Thesis, University of Bath, United Kingdom.

STEP Tools Inc., 2001a, "What is STEP ?",http://www.steptools.com/library/about_step.html.

STEP Tools Inc., 2001b, "The ISO STEP standards", http://www.steptools.com/library/standard/step.html.

Stirk, C.W., Liu, Q. & Ball, M., 1999, "Manufacturing cost analysis of integrated photonic packages", *Proceedings of the 1999 OptoelectronicIntegrated Circuits and Packaging III.*

Subrahmanian, E., Konda, S.L., Levy, S.N., Reich, Y., Westerberg, A.W. & Monarch, I., 1993, "Equations Aren't Enough: informal Modelling in Design", *In Artificial intelligence in Engineering Design, Analysis and manufacturing* (AI EDAM), Vol 7, Part 4, pp. 257-274.

Taylor, J.W. & Talbot, A.C., 1998, "Key system approach to the concurrent definition, verification, and manufacture of main shafts for gas turbine aero engines", *Aerospace Transmission Systems – Concurrent Design and Manufacture,* Institution of Mechanical Engineers Seminar Publication, ISBN 1-86058-161-7.

The Concise Oxford Dictionary of Current English, 1990, Clarendon Press, Oxford, UK, ISBN 0-1986-1319-9.

Theobald, G., 1995, "*Modelling standard components for engineering systems*", Ph.D. Thesis, University of Bath, UK.

The Mathworks Inc., 2001, "The MathWorks V.12", http://www.mathworks.com.

Thoma, J.U., 1975, "*Introduction to Bond Graphs and their Applications*", UK: Pergamon Press, USA, ISBN 0-08018-882-6.

TRIZ Journal, 2001, "The TRIZ journal", http://www.triz-journal.com.

UGS, 2001, WAVE, "Homepage for UGS and Unigraphics CAE system", http://www.ugsolutions.com/products/wave.htm.

Ullman, D.G., 1993, "The Evolution of Function and Behaviour during Mechanical Design", *American Society of Mechanical Engineer, Design Theory and Methodology,* DE-Vol. 53, pp. 91-103.

Ullman, D.G., 1992, "*The Mechanical Design Process*", McGraw-Hill, USA, ISBN 0-07-065739-4.

Ulrich, K.T. & Eppinger, S.D., 2000, "*Product Design and Development*", Second Edition, Irwin McGraw-Hill, New York, USA, ISBN 0-07-229647-X.

Ulrich, K.T., 1995, "*Product Design and Development*", McGraw-Hill, Inc., New York, USA, ISBN 0-07-065811-0.

Valasek, M. & Zdrahal, Z., 1997, "Experiments with Applying Knowledge Based Techniques to Parametric Design", *Proceedings of the International Conference on Engineering Design* ICED 97, pp. 277-280.

VDI 2221, 1986, "*Systematic Approach to the Design of Technical Systems and Products*", VDI Society for Product Development, Design and Marketing.

Vogwell, J., Culley, S.J., Theobald, G. & Ajderian, N., 1993, "Design and optimisation of engineering assemblies using standard components", ACME Final Review Report, GR/F 27734, University of Bath, UK.

Vogwell, J. & Culley, S.J., 1991, "A strategy for selecting engineering components", *Proceedings Institution of Mechanical Engineers*, Part B: Journal of Engineering Manufacture, Vol. 205, pp. 11-17.

Vonderembse, M.A. & White, G.P., 1991, "*Operations Management: Concepts, Methods and Strategies*", Second Edition, West Publishing Company, New York, USA, ISBN 0-314-78148-X.

Wallace, A.P., 1995, "*The Modelling of Engineering Assemblies Based on Standard Catalogue Components*", M.Phil., University of Bath, UK.

Walsh, G.R., 1975, "*Methods of Optimization*", John-Wiley & Sons, London, UK, ISBN 0-471-91924-1.

Ward, A.C. & Seering, W.P., 1989, "The performance of a mechanical design compiler", American Society of Mechanical Engineers, Design Engineering, Vol. 17, pp. 89-97.

Ward, A.C. & Seering, W.P., 1990, "The performance of a mechanical design compiler", *Proceedings of the1990 ASME International Computers in Engineering Conference.* pp. 187-192.

Webber, S.J., 1994, "*Techniques for the analysis of search results in computer aided selection systems*", Ph.D. Thesis, University of Bath, UK.

Whitney, D.E., 1990, "Designing the Design Process", *Research in Engineering Design*, Springer-Verlag New York Inc, pp. 3-13.

Womack, J.P., Jones, D.T. & Roos, D., 1990, "*The Machine that Changed the World*", Maxwell Macmillan International, New York, ISBN 0-89256-350-8.

Woolfson, M.M. & Pert, G.J., 1999, "*Introduction to Computer Simulation*", Oxford University Press, UK, ISBN 0-19850-423-3.

Yang, Y.N., Parsaei, H.R., Hamid, R., Leep, H.R. & Wong, J.P., 1998, "Manufacturing cost-estimating system using activity-based costing", *International Journal of Flexible Automation and Integrated Manufacturing*, Vol. 6, No. 3, pp. 223-243.

Zeid, I., 1991, "CAD/CAM Theory and Practice", McGraw-Hill Series in Mechanical Engineering, New York, USA, ISBN 0-07-072857-7.

Appendix

Electronic representations

1. Shaft representation in C
2. Gear representation in Microsoft Excel
3. Bearing representation in Microsoft Access
4. Chain drive representation in Visual Basic
5. Keyway representation in BASIC (ActiveX component)
6. Mount representation in BASIC (ActiveX component)
7. Load representation in Microsoft Excel
8. Cam representation in Microsoft Excel and SolidWorks

Shaft representation

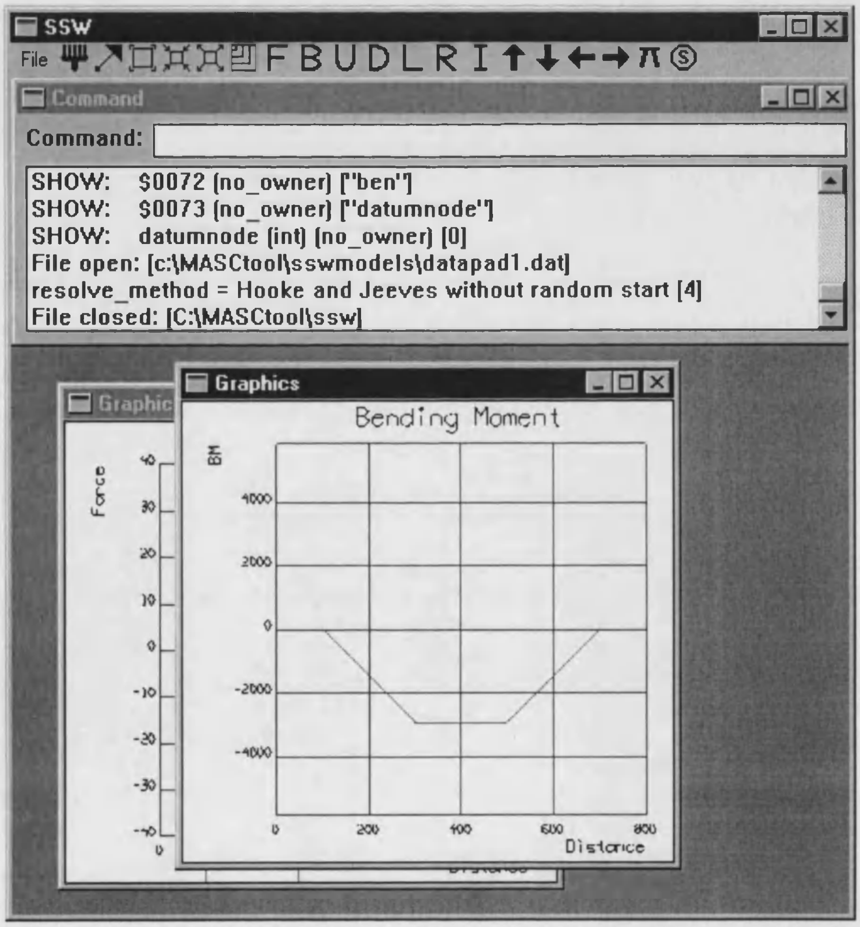
Software environment: C

Component attributes:

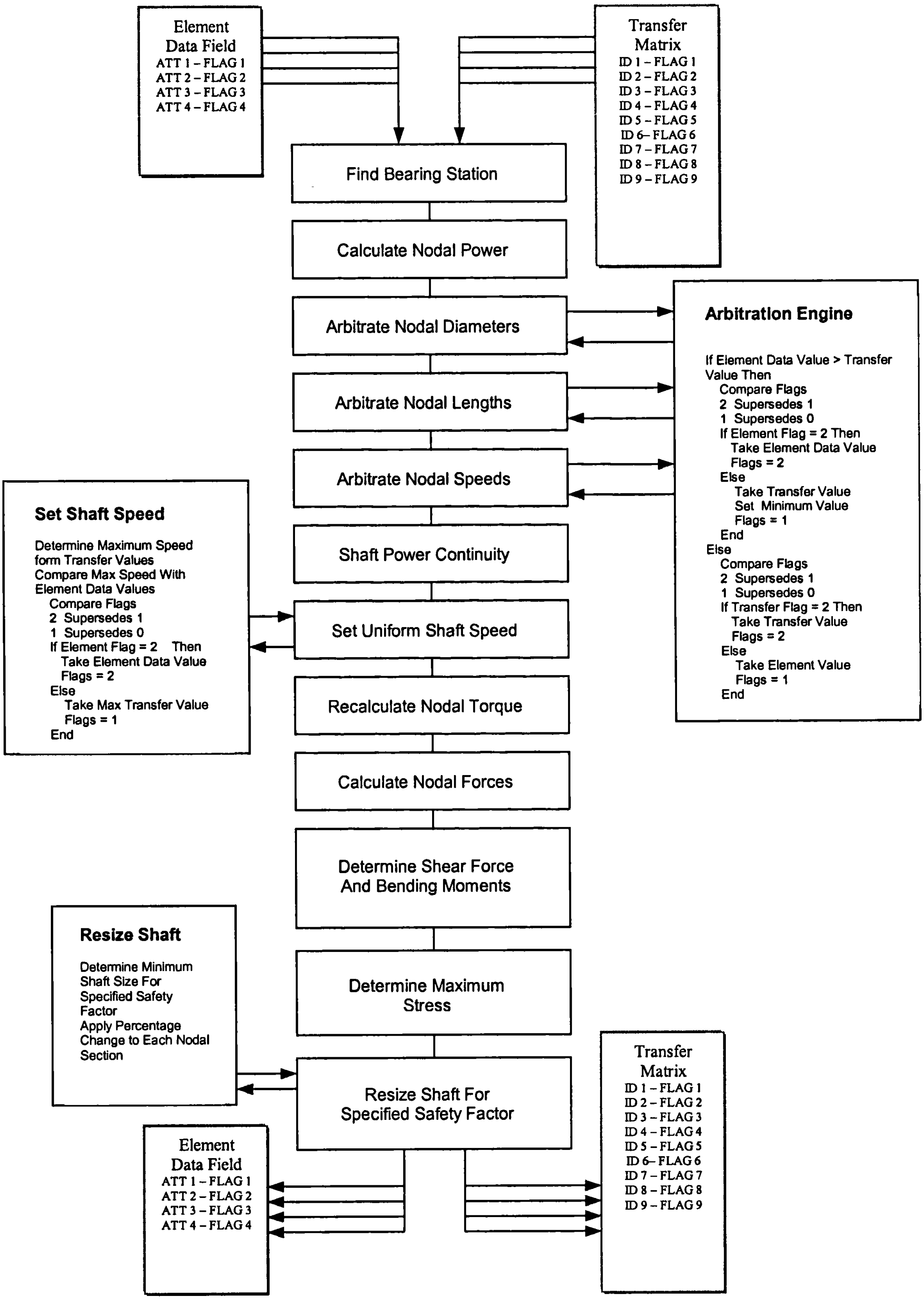
Number of attributes: 6

Parameter	Value	Units	Flag
power	2	kW	0
node diameter	20	mm	0
node length	30	mm	0
speed	10	rpm	0
material	0.207	0.207	0
Safety factor	2	n/a	0

Screen shot:



Selection/specification process:



Gear representation

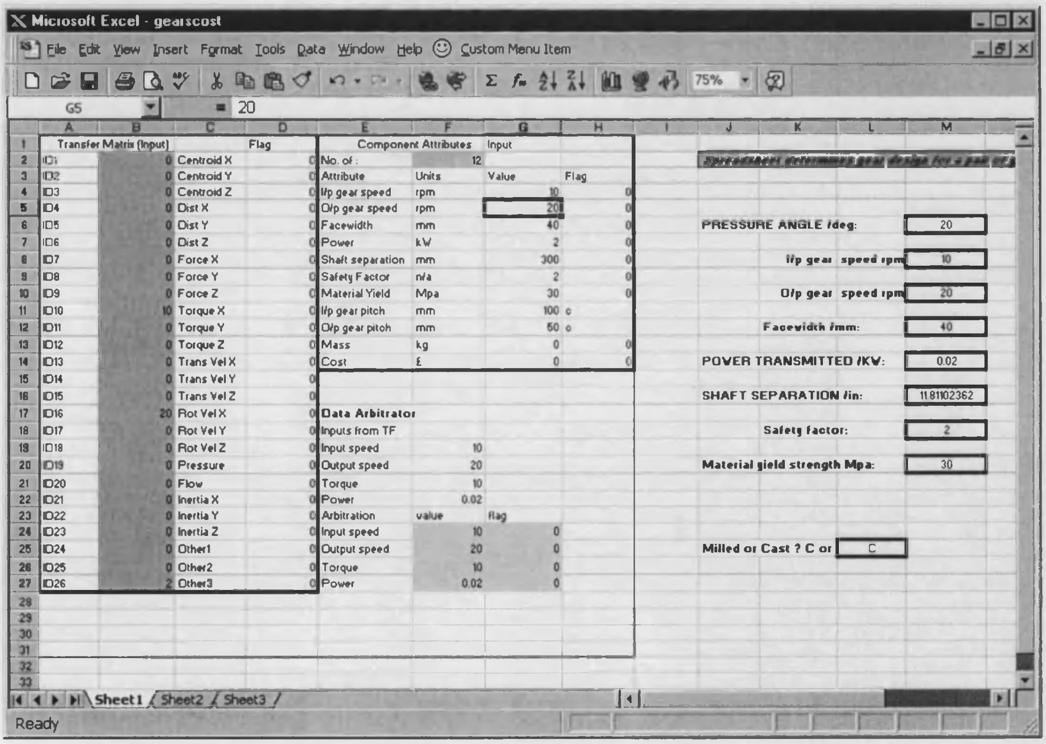
Software environment: Microsoft Excel Spreadsheet

Component attributes:

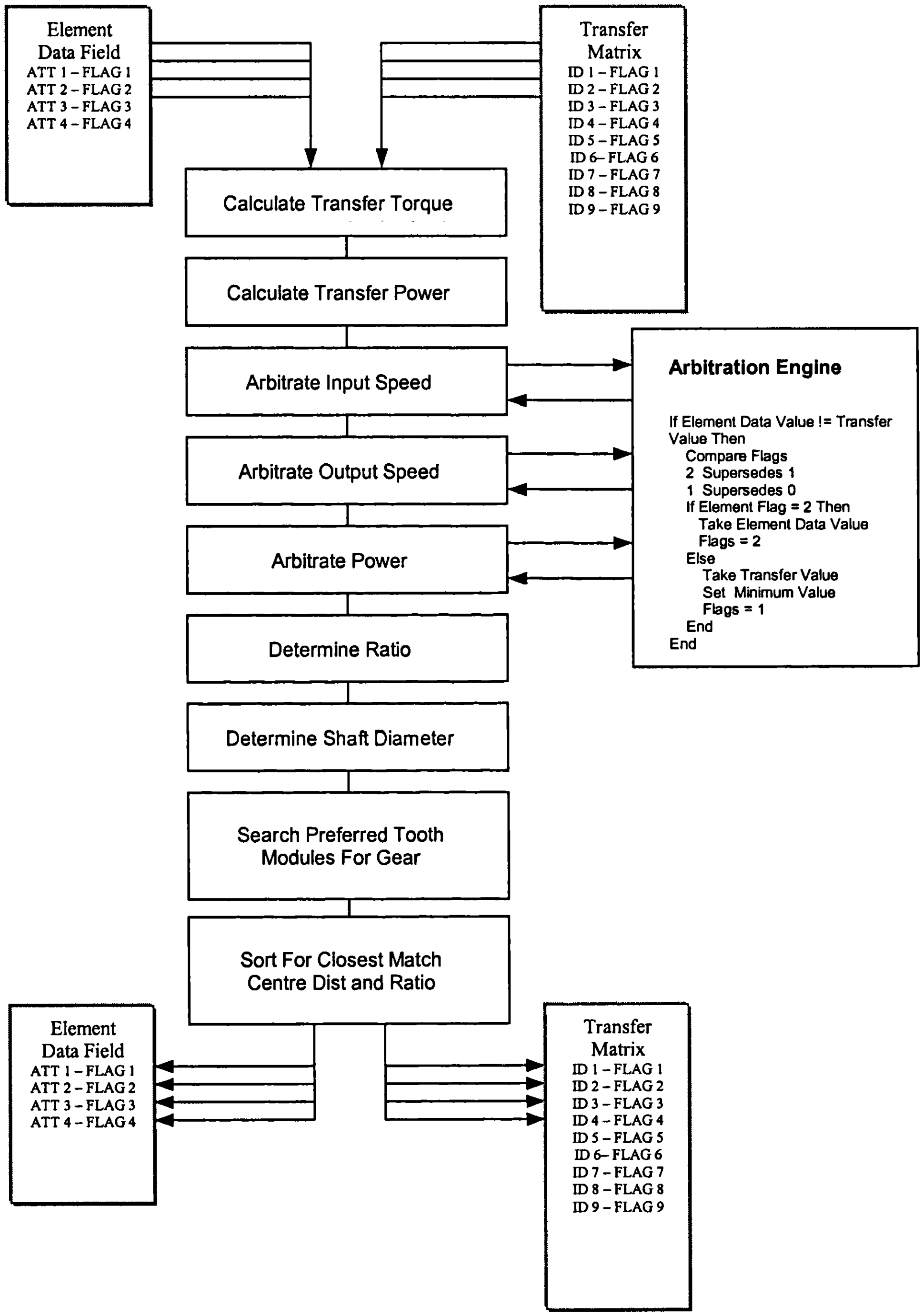
Number of attributes: 9

Parameter	Units	Value	Flag
I/p gear speed	rpm	10	0
O/p gear speed	rpm	21.20	0
Face width	mm	40	1
Power	kW	0.020944	0
Shaft separation	mm	97.5	1
Safety Factor	n/a	2	1
Material Yield	Mpa	30	1
I/p gear pitch	mm	132.5	c
O/p gear pitch	mm	62.5	c

Screen shot:



Selection/specification process:



Bearing representation

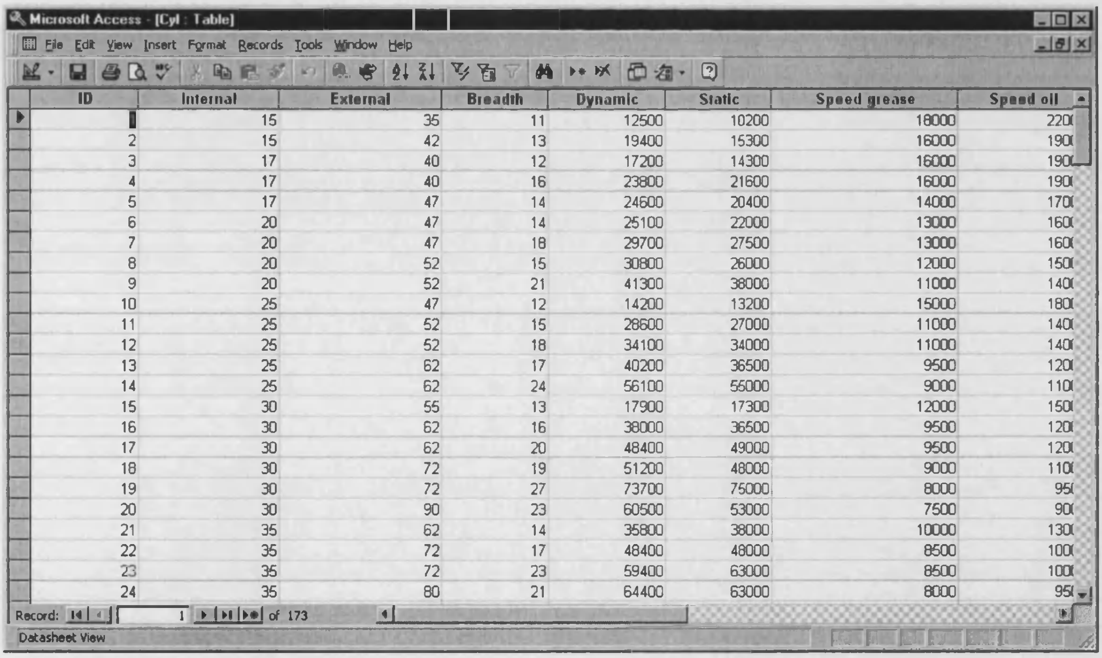
Software environment: Access database

Component attributes:

Number of attributes: 8

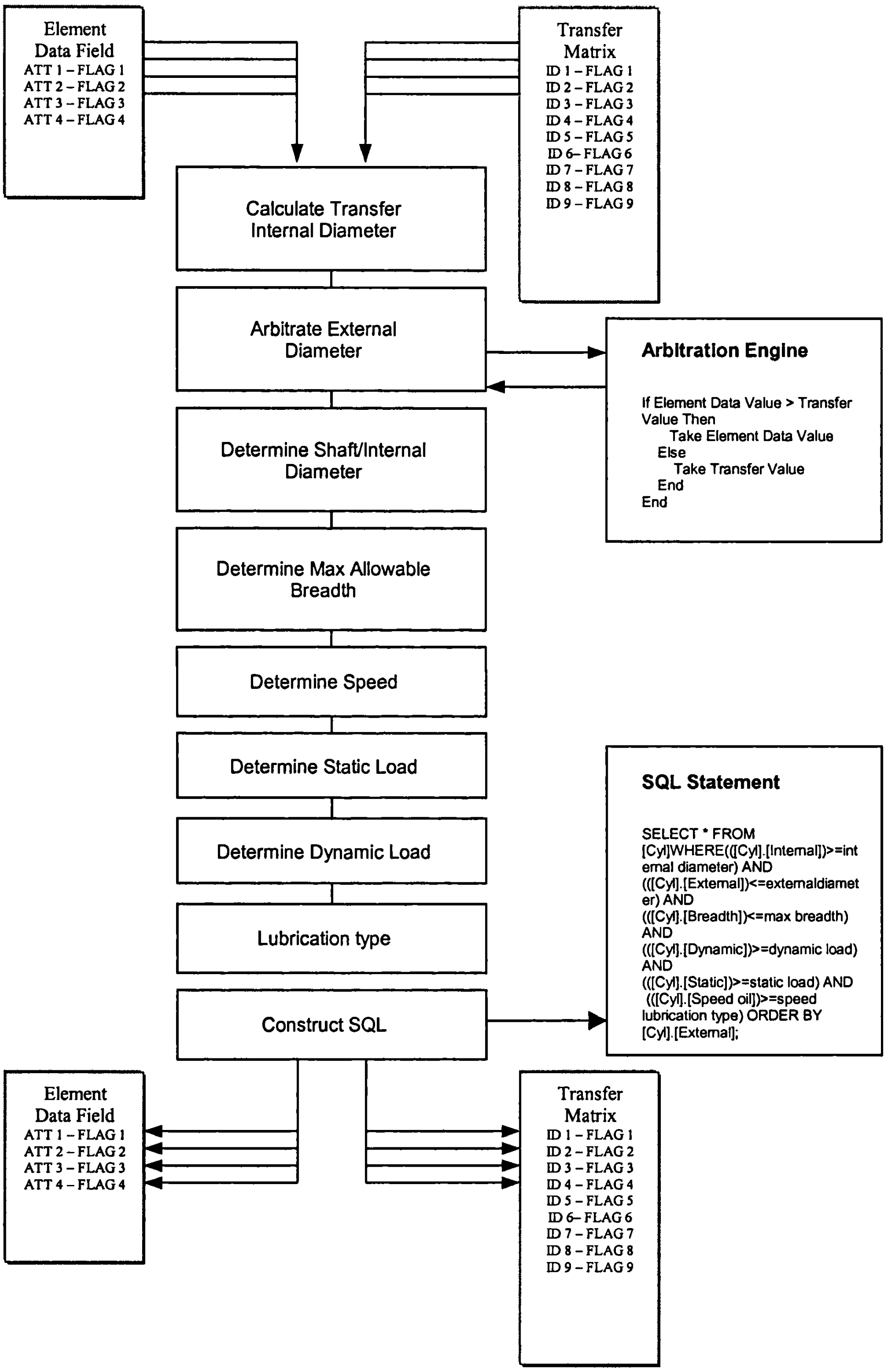
Parameter	Value	Units	Flag
Internal diameter	1	mm	0
External diameter	1000	mm	0
breadth	200	mm	0
Dynamic load	1	N	0
Static load	1	N	0
Speed rating	1	rpm	0
lubrication	oil	oil/grease	0
life	1000	10^6	0

Screen shot:



ID	Internal	External	Breadth	Dynamic	Static	Speed grease	Speed oil
1	15	35	11	12500	10200	18000	220
2	15	42	13	19400	15300	16000	190
3	17	40	12	17200	14300	16000	190
4	17	40	16	23800	21600	16000	190
5	17	47	14	24600	20400	14000	170
6	20	47	14	25100	22000	13000	160
7	20	47	18	29700	27500	13000	160
8	20	52	15	30800	26000	12000	150
9	20	52	21	41300	38000	11000	140
10	25	47	12	14200	13200	15000	180
11	25	52	15	28600	27000	11000	140
12	25	52	18	34100	34000	11000	140
13	25	62	17	40200	36500	9500	120
14	25	62	24	56100	55000	9000	110
15	30	55	13	17900	17300	12000	150
16	30	62	16	38000	36500	9500	120
17	30	62	20	48400	49000	9500	120
18	30	72	19	51200	48000	9000	110
19	30	72	27	73700	75000	8000	95
20	30	90	23	60500	53000	7500	90
21	35	62	14	35800	38000	10000	130
22	35	72	17	48400	48000	8500	100
23	35	72	23	59400	63000	8500	100
24	35	80	21	64400	63000	8000	95

Selection/specification process:



Chain drive representation

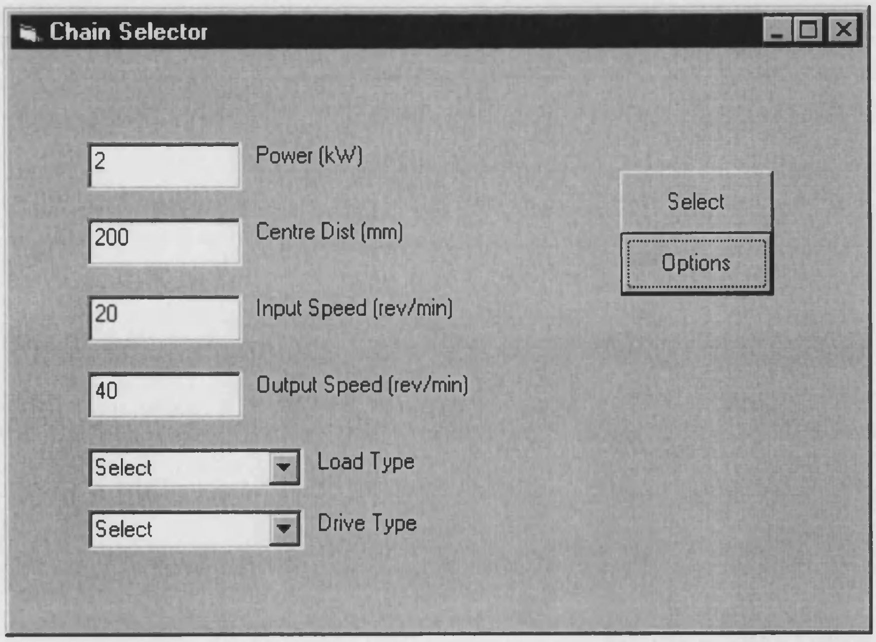
Software environment: Visual Basic

Component attributes:

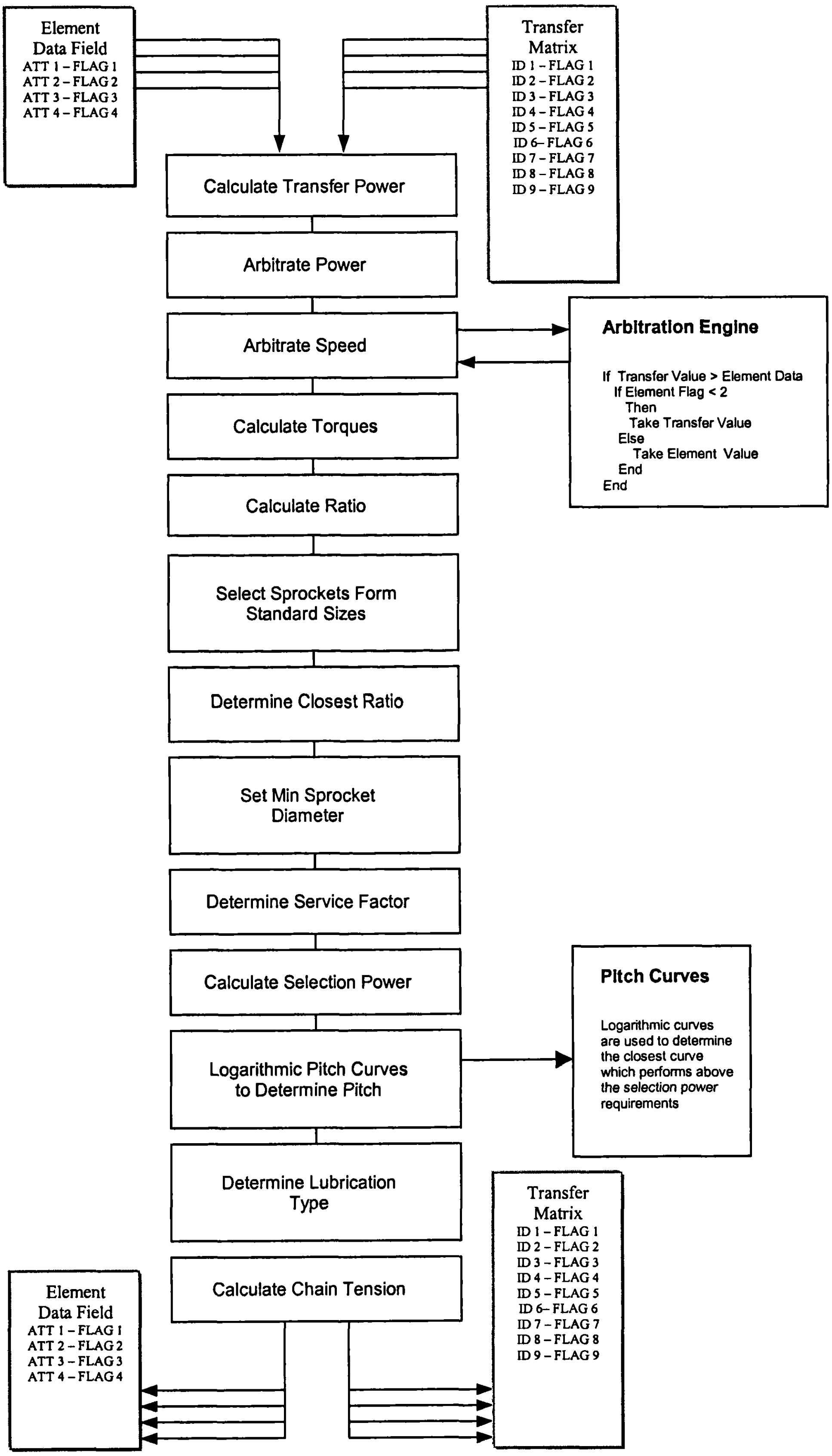
Number of attributes: 13

Parameter	Value	Units	Flag
power	2	kW	0
centre distance	100	mm	0
input speed	10	rpm	0
output speed	20	rpm	0
load type	smooth	moderate/heavy	0
drive type	steady	medium/heavy	0
ratio	n/a	n/a	0
no of strands	n/a	n/a	0
pitch	n/a	mm	0
force	n/a	N	0
chain length	n/a	mm	0
no of links	n/a	n/a	0
lubrication type	n/a	n/a	0

Screen shot:



Selection/specification process:



Keyway representation

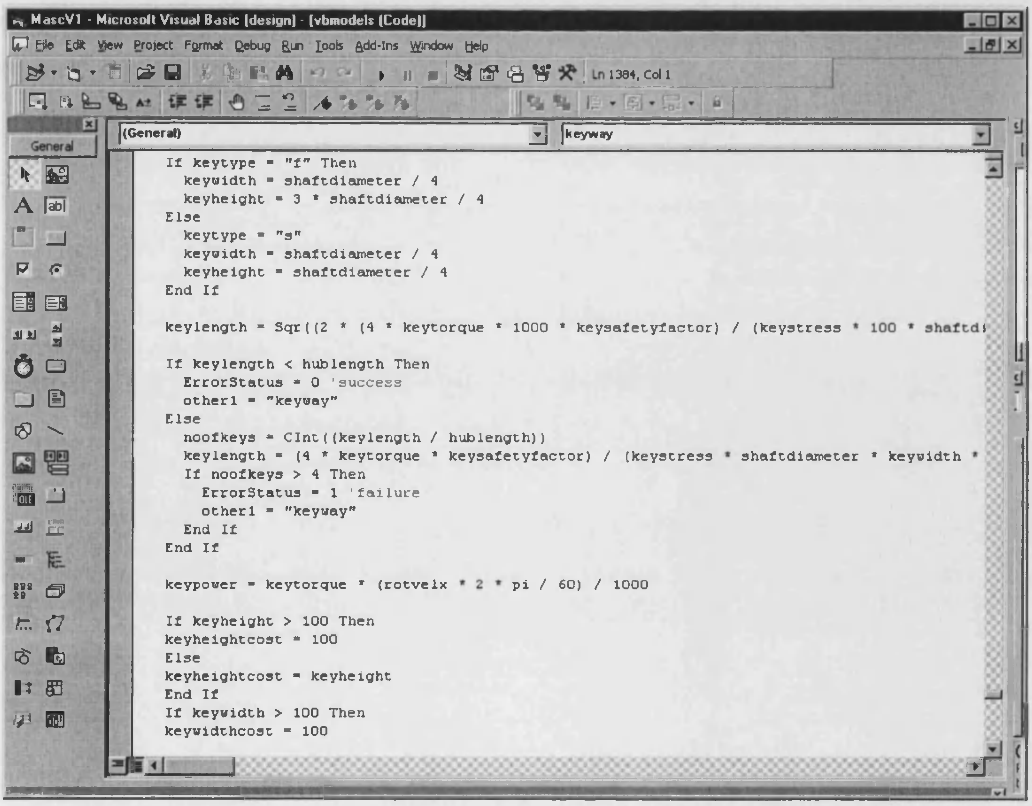
Software environment: BASIC (ActiveX component)

Component attributes:

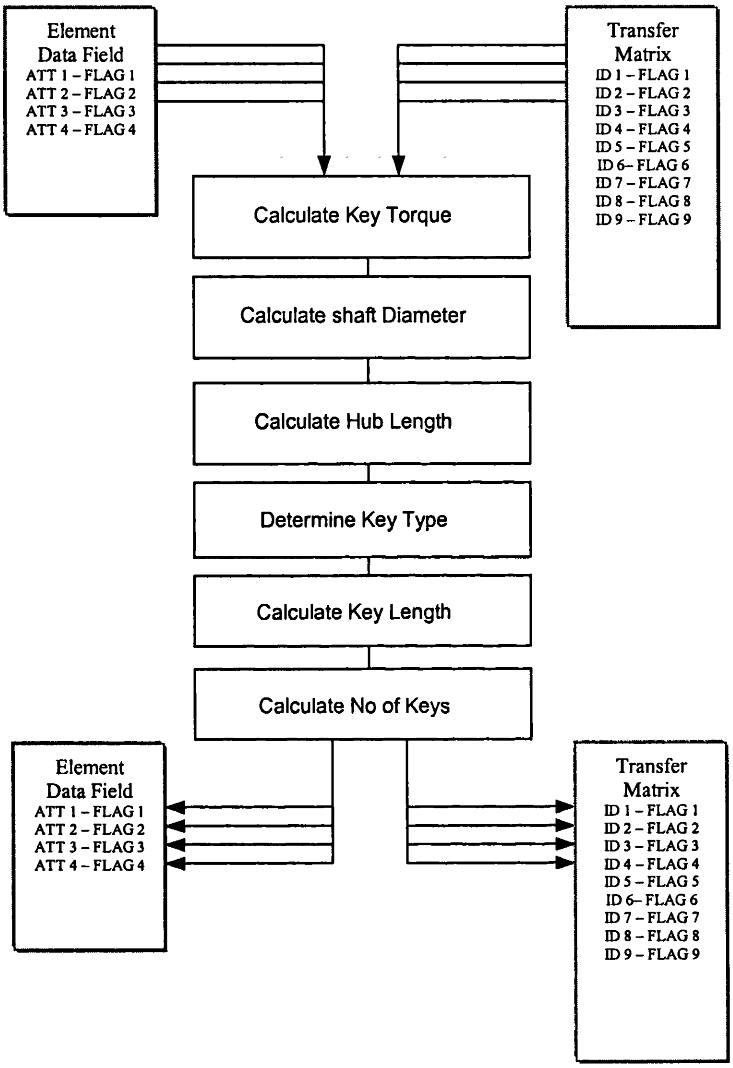
Number of attributes: 13

Parameter	Value	Units	Flag
no of keys	1	n/a	0
key type	s	s or f	0
width	7	mm	0
height	10	mm	0
length	20	mm	0
shaft diameter	20	mm	0
hub length	30	mm	0
torque	2	Nm	0
speed	10	rpm	0
power	2	kW	0
stress	100	Mpa	0
shear stress	10	Mpa	0
safety factor	2	n/a	0

Screen shot:



Selection/specification process:



Mount representation

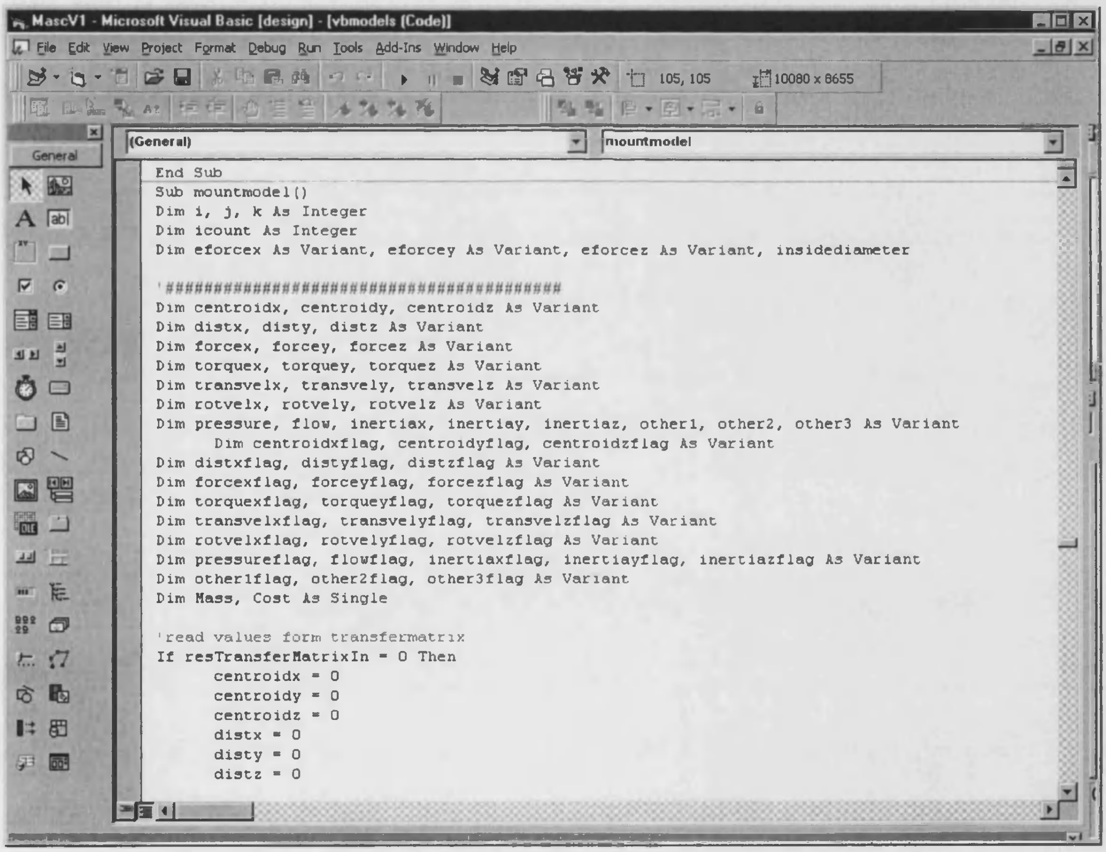
Software environment: BASIC (ActiveX component)

Component attributes:

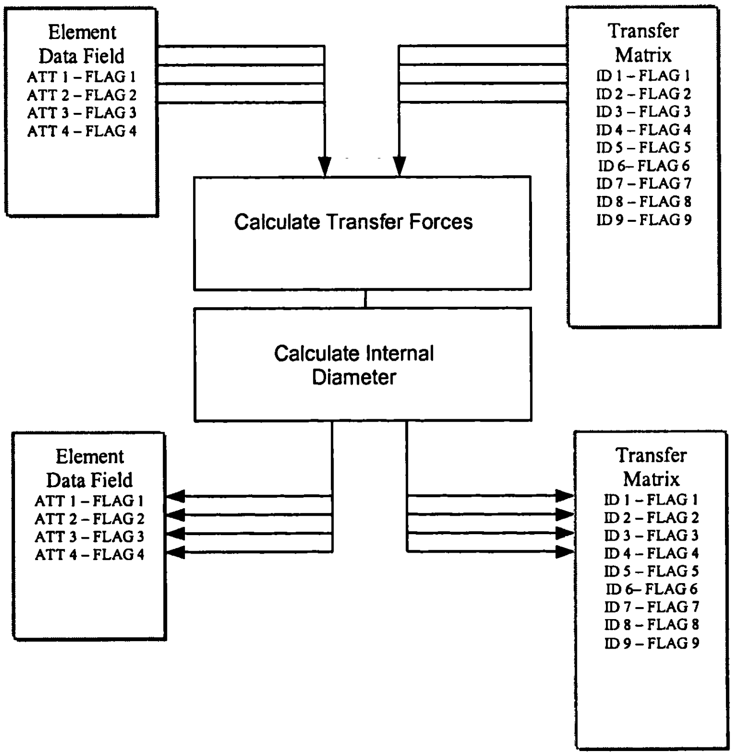
Number of attributes: 4

Parameter	Value	Units	Flag
inside diameter	0	N	0
force x	0	N	0
force y	0	N	0
force z	0	N	0

Screen shot:



Selection/specification process:



Load representation

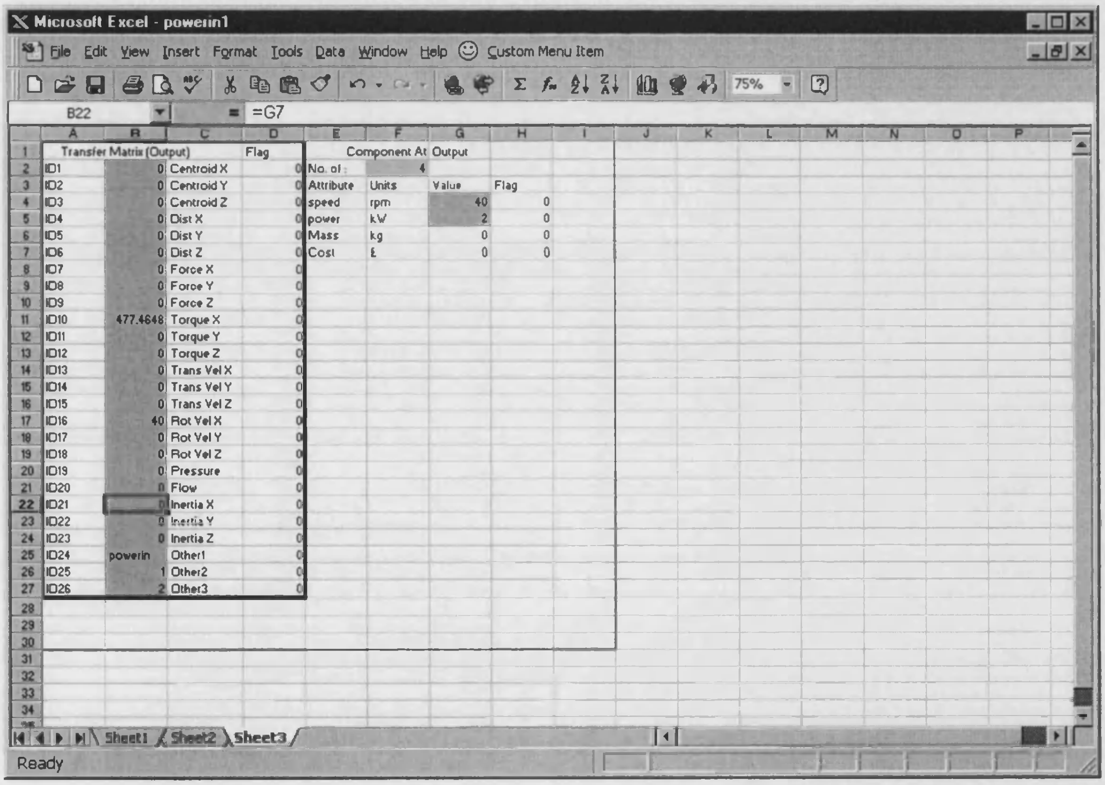
Software environment: Microsoft Excel Spreadsheet

Component attributes:

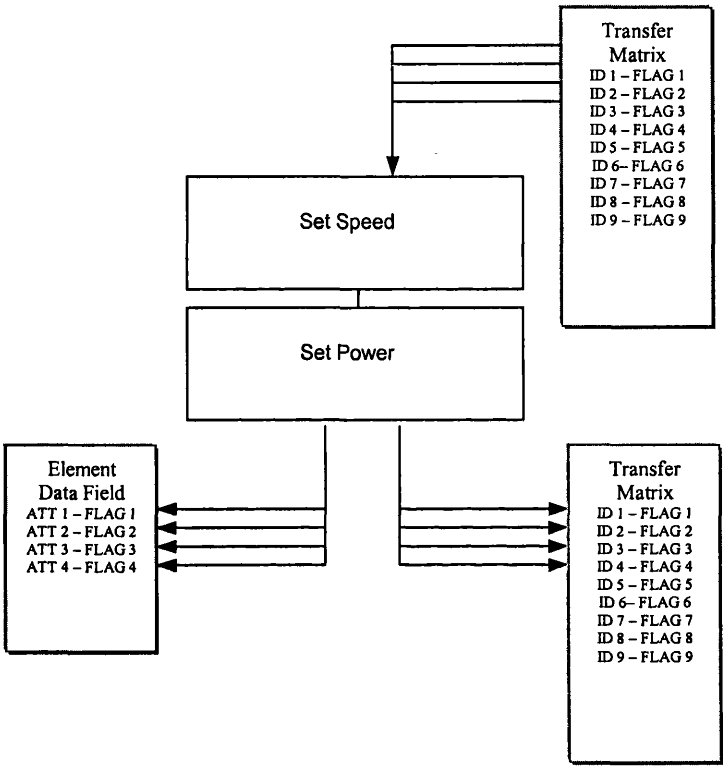
Number of attributes: 2

Parameter	Value	Units	Flag
speed	40	rpm	0
power	10	kW	0

Screen shot:



Selection/specification process:



Selection/specification process: